Title: Accounting company web application

Candidate name: Thanyaphon Limmanee (IngIng)

Word count: 1988 words

Criteria A: Planning

Defining existing problem

Ms. Pattaraporn is the owner of an accounting firm that manages and consults on the accounting aspect of businesses. As a result, she dealt with a massive amount of data and papers from many businesses that span several years. Currently, she is using a paper-based management of her documents which is prone to misplacement and disorganisation given the large amount of files she manages as mentioned in interview 1 in the appendix. She also said that some businesses her firm handles may also have a change in their company registration during the year and when it comes to the closing financial statement period, these registrations need to be addressed due to the large amount of business she has to handle, some of them got missed out.

She is eager to transition to an online document storage solution to address these concerns. However, the existing accounting software and programs are mostly built for small enterprises wanting to handle their finances, making them unsuitable for the demands of her accounting firm. The company also has employees who assist with different aspects of accountancy and have different levels of access to the documents for the data security aspect of the company's integrity so she requests software that provides different access to each user. Consequently, she wants software that can store the documents, perform basic taxing calculations, and provide her employees with separate access.

Word count: 230 words

The rationale for the proposed solution

Because this program needs to be accessed by different users from different platforms, I decided to use a web application to program this database software. The web application allows users to access different devices as they all can access through the website which fits the working environment in Ms. Pattaraporn's accounting firms where every employee has a computer to work on. It also allows all user to access the same version of the application, which doesn’t need installation which means there will be no need to update the application themselves and it saves storage space in their device which can be used to store other important documents privately. However, the web application could be easily accessed with no vetting or approval needed meaning it will have lower data security; to handle this issue, I will add a user login function to improve the security and prevent anyone who doesn’t have an account from accessing the web application.

I will be using Java to program this web application through the integrated development environment Apache Netbeans which I studied in class along with utilising Javascript and HTML for the interface for a number of reasons. Firstly, it is a platform-independent language which means that the web application code can be developed once and have it run across multiple platforms without having to redo it for each platform; this will save time and cost for the development. Secondly, it is object-oriented programming for higher productivity, lower data redundancy, and higher security as I can implement features like inheritance and private classes. Lastly, the availability of the application programming interface of Java is very high such as the database connection JDBC that allows me to easily connect to SQL database[1] which I will be utilising as it is simpler whereas in Hibernate it is more complex and takes more time to learn[2], so considering the amount of time given to complete this project, JDBC is more suitable.

Word count: 323 words

[1] "Introduction to JDBC | Baeldung." 6 May. 2023, https://www.baeldung.com/java-jdbc. Accessed 7 Jan. 2024.

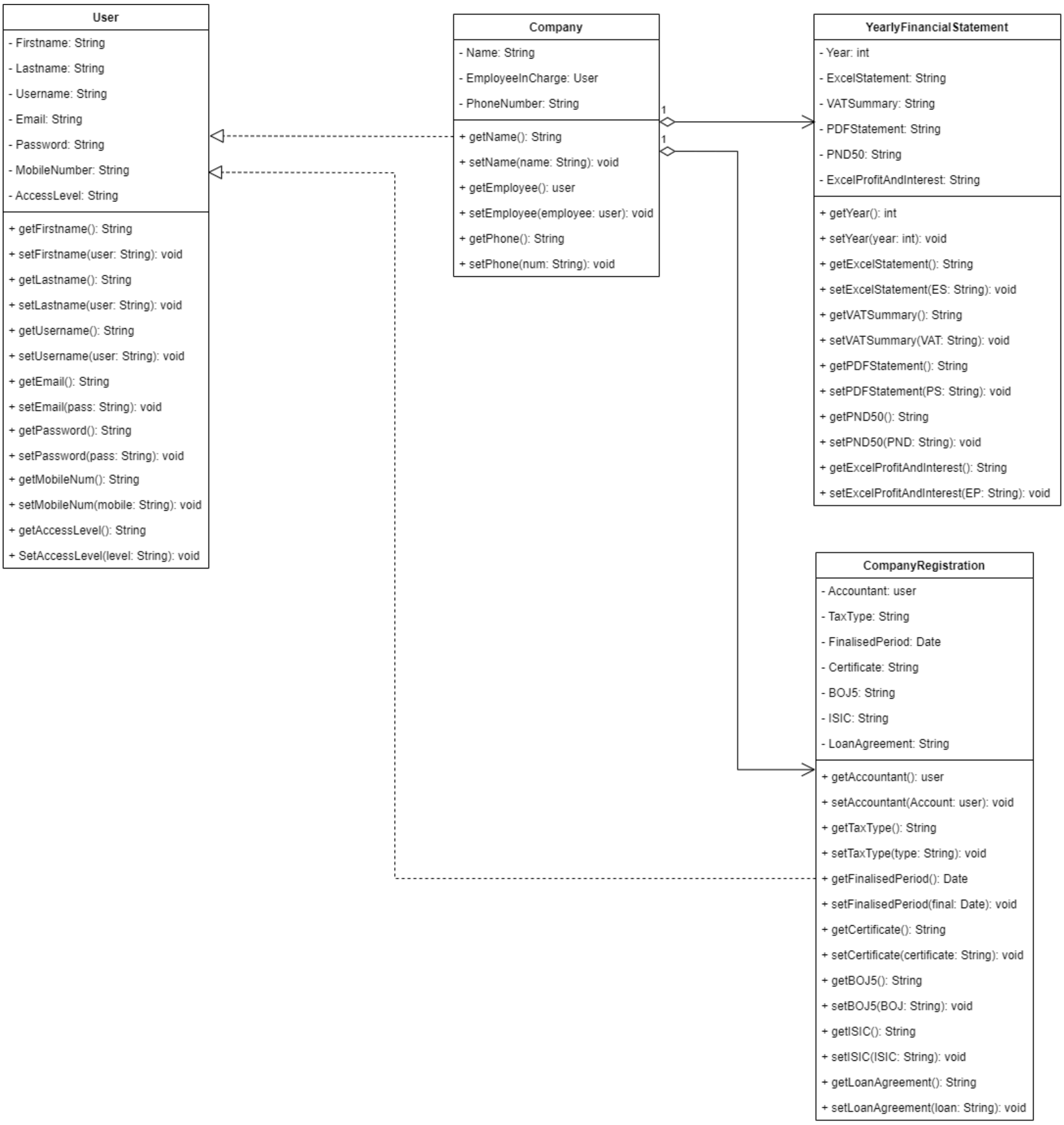
[2] "Hibernate vs JDBC: Which to Choose? - DevTeam.Space." https://www.devteam.space/blog/hibernate-vs-jdbc-which-to-choose/. Accessed 7 Jan. 2024.

Success criteria

- Login page for different types of access
    - Administrator - can edit, delete, and access everything
    - Manager - can add and access but not delete
    - Employees - can only access
- Company list interface
- Sort all company registrations by date
- Past company registration list for each company
- The financial statement list for each company
- Sort financial statement list by year
- Add and remove a company from the list
- Add and remove a financial statement from the list
- Add and remove company registration from the list
- Be able to upload different types of files and store it
- Be able to download the stored different types of files

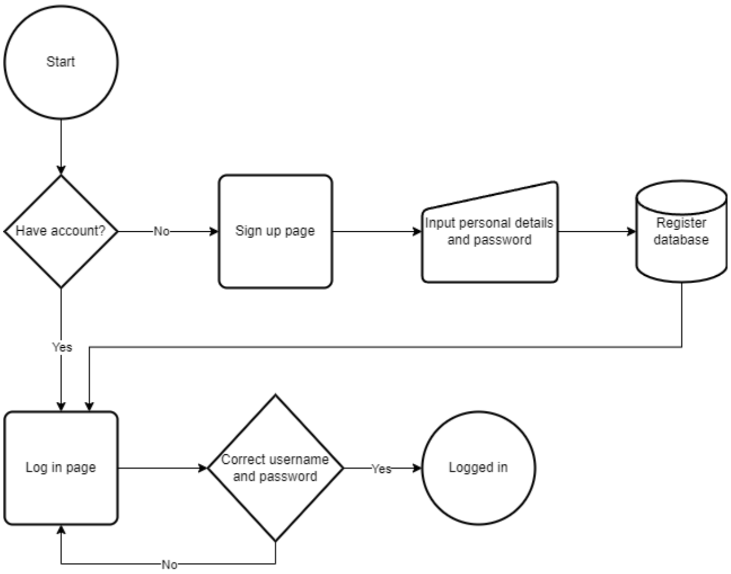
Criteria B: Design Overview

OOP class UML diagram

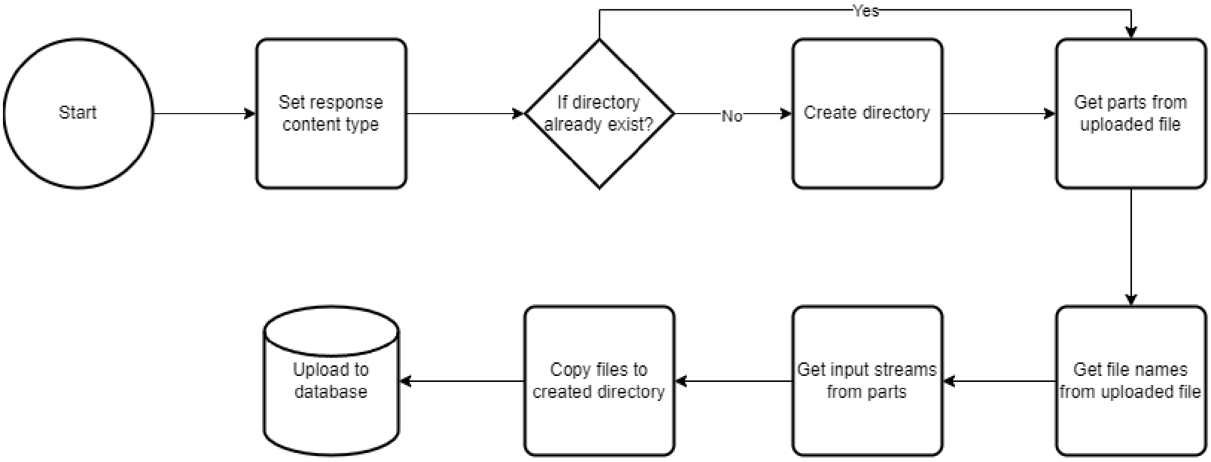

User
- Firstname: String
- Lastname: String
- Username: String
- Email: String
- Password: String
- MobileNumber: String
- AccessLevel: String
+ getFirstname(): String
+ setFirstname(user: String): void
+ getLastname(): String
+ setLastname(user: String): void
+ getUsername(): String
+ setUsername(user: String): void
+ getEmail(): String
+ setEmail(pass: String): void
+ getPassword(): String
+ setPassword(pass: String): void
+ getMobileNum(): String
+ setMobileNum(mobile: String): void
+ getAccessLevel(): String
+ SetAccessLevel(level: String): void
Company
- Name: String
- EmployeeInCharge: User
- PhoneNumber: String
+ getName(): String
+ setName(name: String): void
+ getEmployee(): user
+ setEmployee(employee: user): void
+ getPhone(): String
+ setPhone(num: String): void
1
1
YearlyFinancialStatement
- Year: int
- ExcelStatement: String
- VATSummary: String
- PDFStatement: String
- PND50: String
- ExcelProfitAndInterest: String
+ getYear(): int
+ setYear(year: int): void
+ getExcelStatement(): String
+ setExcelStatement(ES: String): void
+ getVATSummary(): String
+ setVATSummary(VAT: String): void
+ getPDFStatement(): String
+ setPDFStatement(PS: String): void
+ getPND50(): String
+ setPND50(PND: String): void
+ getExcelProfitAndInterest(): String
+ setExcelProfitAndInterest(EP: String): void
CompanyRegistration
- Accountant: user
- TaxType: String
- FinalisedPeriod: Date
- Certificate: String
- BOJ5: String
- ISIC: String
- LoanAgreement: String
+ getAccountant(): user
+ setAccountant(Account: user): void
+ getTaxType(): String
+ setTaxType(type: String): void
+ getFinalisedPeriod(): Date
+ setFinalisedPeriod(final: Date): void
+ getCertificate(): String
+ setCertificate(certificate: String): void
+ getBOJ5(): String
+ setBOJ5(BOJ: String): void
+ getISIC(): String
+ setISIC(ISIC: String): void
+ getLoanAgreement(): String
+ setLoanAgreement(loan: String): void

Web page flow chart

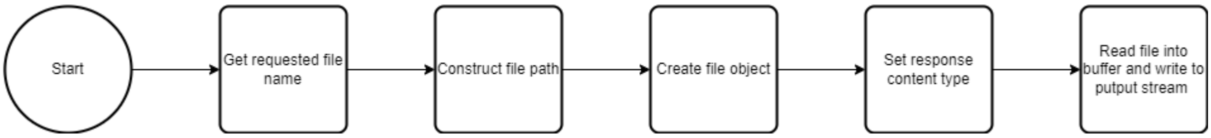
Log in & Sign up page



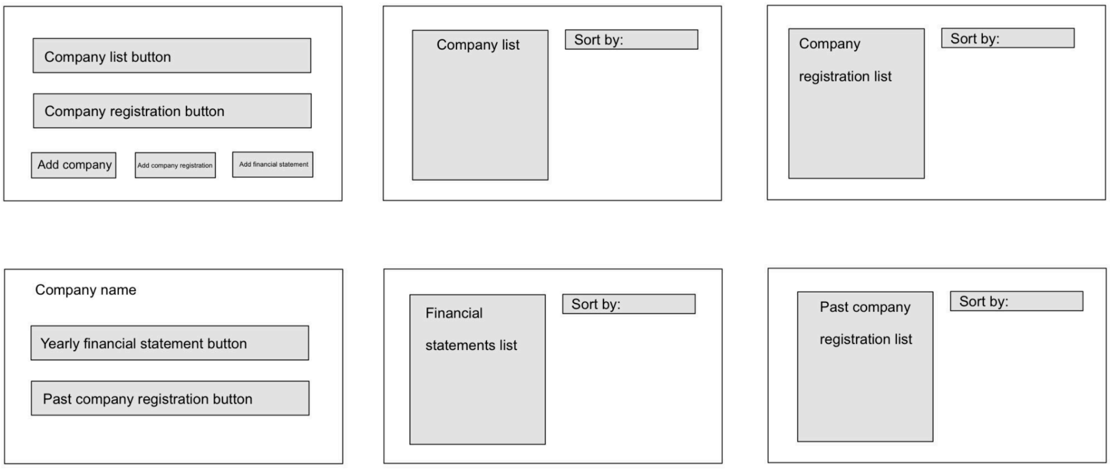
Upload files


Yes
Start
Set response content type
If directory already exist?
No
Create directory
Get parts from uploaded file
Get file names from uploaded file
Get input streams from parts
Copy files to created directory
Upload to database

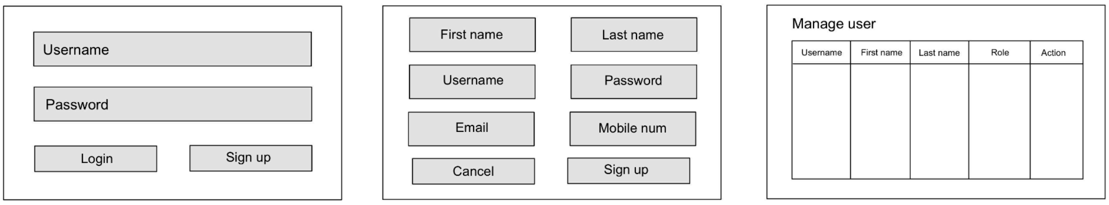
Download files


Start
Get requested file name
Construct file path
Create file object
Set response content type
Read file into buffer and write to putput stream

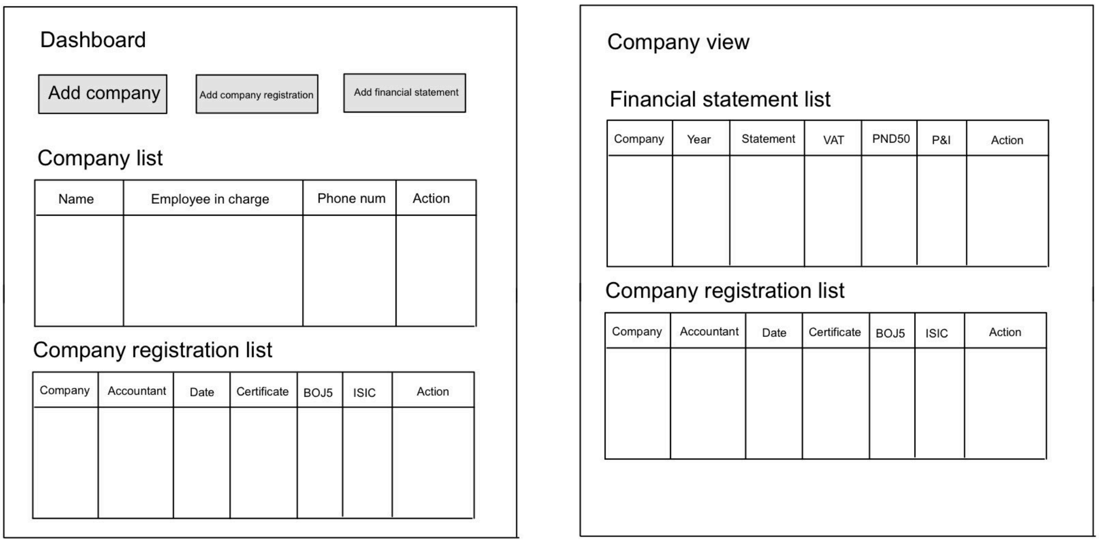
Interface design 1 - Main functions


Company list button
Company registration button
Add company
Add company registration
Add financial statement
Company list
Sort by:
Company registration list
Sort by:
Company name
Yearly financial statement button
Past company registration button
Financial statements list
Sort by:
Past company registration list
Sort by:

Interface design 2 - Login, Signup & Manager User



Interface design 3 - Main functions redesign



Database Table

company

Field name	Data type	Description	Sample data
name	varchar	Field for the name of registered company.	“Trial 1”
employeeInCharge	varchar	Field for employee in the financial company which is in charge of managing specific business (foreign key to username in register table)	“Joe”
phoneNumber	varchar	Field for contact phone number for each company	“1234567890”
ID	int	Unique index for each company identification (primary key)	<auto incremented>

company registration

Field name	Data type	Description	Sample data
company	varchar	Field for the company name the company registration belongs to (foreign key to name in company table)	“Trial 1”
accountant	varchar	Field for assigned accountant in each company registration (foreign key to username in register table)	“Joe”
date	date	Field for date registered for each company registration	“2024-02-15”
certificate	blob	Field for input evidence certificate file	<JPEG file>
boj5	blob	Field for input boj5 file	<PDF file>
isic	blob	Field for input isic file	<PDF file>
loanAgreement	blob	Field for input loan agreement file	<JPEG file>
ID	int	Unique index for each company registration identification (primary key)	<auto incremented>

financial statement

Field name	Data type	Description	Sample data
company	varchar	Field for the company name the financial statement belongs to (foreign key to name in company table)	“Trial 1”
year	int	Field for year regstered of each financial statement	2024
excelStatement	blob	Field for input financial statement in excel format	<Excel file>
pdfStatement	blob	Field for input financial statement in PDF format	<PDF file>
vatSummary	blob	Field for input VAT summary file	<PDF file>
pnd50	blob	Field for input pnd50 file	<PDF file>
excelProfitAndInterest	blob	Field for input profit and interest file ini excel format	<Excel file>
ID	int	Unique index for each financial statement identification (primary key)	<auto incremented>

register

Field name	Data type	Description	Sample data
firstname	varchar	Field for first name of registered account	“Joe”
lastname	varchar	Field for last name of registered account	“Smith”
username	varchar	Field for username of registered account and must be unique (primary key)	“Joe”
password	varchar	Field for password of registered account use fo authentication on log in process	“12345”
email	varchar	Field for contact information of registered account via email	“123@gmail.com”
mobileNum	varchar	Field for contact information of registered account via	“1234567890”

		mobile number	
access	varchar	Field for identify the access level of each account. Admin can edit, delete, and access everything, manager can add and access but not delete, and employee can only access.	“manager”

Test plan

Test type	Nature of test	Example
User with different access levels can access different parts and function of the program	Log in with account which have different access level	Login with “admin” - Can add, edit, delete, and access everything Login with “manager” - Can add and access but not delete Login with “employee” - Can only accecss
Company list pop up when login	Check the interface that appears after login that it contain no graphical and functional error	User login ad can view the company table list and access the right function
Company registration list can be sorted by date	Click the sort button on the company registration table in date column	The list can be sorted from earliest to latest
Show company registration for each specific company	Click on view company detail in the list and check if there is any graphical or functional error	Company registration list for a single company is displayed when clicked on that company detail button
Show financial statement for each specific company	Click on view company detail in the list and check if there is any graphical or functional error	Financial statement list for a single company is displayed when clicked on that company detail button
Financial statement list can be sorted by year	Click the sort button on the financial statement table in year column	The list can be sorted from earliest to latest
New company can be registered and unused company can be deleted	Add new company and delete unused company	New company registered will be displayed in the table and deleted company will disappears as well as company registration and financial statement for that company
New company registration can be added and unwanted company registration can be deleted	Add new company registration and delete existing company registration	New company registration added will be displayed in the table and deleted company registration will disappears
New financial statement can be added and unwanted financial statement can be	Add new financial statement and delete existing financial statement	New financial statement added will be displayed in the table and deleted

deleted		financial statement will disappears
Can upload and stored different type of documents such as PDF, PNG, Excel	Upload new documents with different types to the site and see if it is stored and displayed for download	Upload sample documents and shown in the table
Can download the stored documents	Download the stored document and check if the content is correct	The document previously uploaded can be download and the data in the document stayed the same

Criteria B: Record of Task

	Candidate:				
Task number	**Planned action**	**Planned outcome**	**Time estimate**	**Target completion date**	**Criterion**
1	Initial discussion with client	figure out their problem and what they want	30 mins	17 October 2023	A
2	Defining problem and proposed solution	deciding what software and coding languages used	1 hrs	22 October 2023	A
3	Complete success criteria	listed success criteria for criterion A	15 mins	22 October 2023	A
4	Purpose idea to advisor	getting the program idea approved	15 mins	24 October 2023	A
5	Beginning design	Start design the functionality and interface of the program	1 hrs	24 October 2023	A
6	Second discussion with client	proposed the designed program to client and ask for their feedback	1 hr	30 October 2023	A
7	Record of task	listing tasks for the whole project and the target completion date	1 hr	10 November 2023	B

8	Basic overview	completing structure and UML diagram	2 hrs	10 November 2023	B
9	Learn basic Web app programming	being able to code basic web app	1 day	12 November 2023	C
10	Create Web app GUI	Completing the GUI of web app	5 days	18 November 2023	C
11	Learn how to use Hibernate ORM	Be able to create and use Hibernate ORM	1 day	20 November 2023	C
12	Create database using Hibernate ORM	successfully create the database and link to the program	2 days	22 November 2023	C
13	Create login system	successfully create login page with multiple access roles	3 days	25 November 2023	C
14	Test the system and database	the user information appeared in the database and identify any bugs	2 days	27 November 2023	C
15	Learn how to upload files to the server	being able to upload pdf, doc, and excel files to the database	2 days	29 November 2023	C
16	Create different class and utilizing OOP	complete overall structure of the program	3 weeks	20 December 2024	C

17	Create tax calculation method	complete method for calculating tax in different scenarios	5 days	25 December 2024	C
18	Use javascript and HTML to work with the web app interface	complete the interface of the program	1 weeks	1 January 2024	C
19	connect everything together	completing the program, tie every class and method together	2 days	3 January 2024	C
20	alpha testing	check the program is working smoothly by using sample data and identify any bugs	2 days	5 January 2024	C, E
21	fix any bugs and software completion	program can run smoothly and ready to be sent to client	2 days	7 January 2024	C
22	Download and instruct client on how to use the program	Teaches client how to access and use the program	1 hr	13 January 2024	E
23	third meeting with client for feedback	gather feedback from the client to modify the program accordingly	1 hr	13 January 2024	E

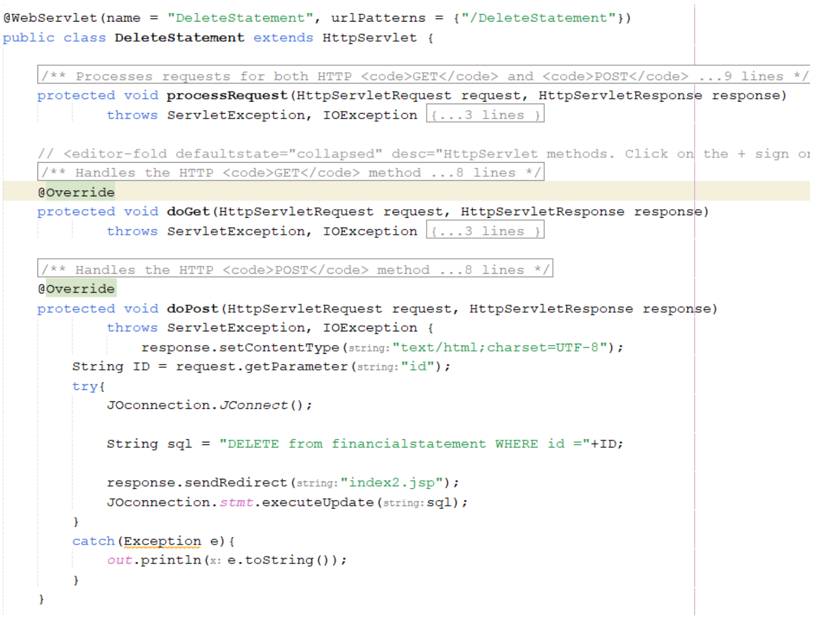
24	fix program for any last changes	finalized the program coding and function according to the feedback	1 week	20 January 2024	C, E
25	Presentation of the program	complete criterion E	1 day	21 January 2024	E

Criteria C: Development

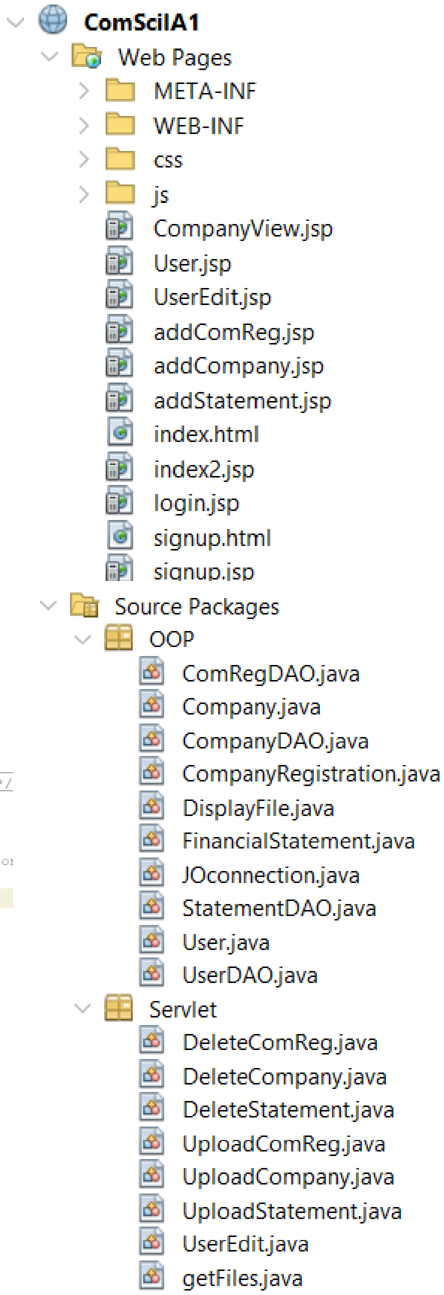
Overall

This project aims to develop a Java-based web application for an accounting company with the utilisation of HTML and CSS for interface presentation and styling. Each file will be separated into 2 main parts; a front-end web pages folder which consists mainly of HTML and CSS with some implementation of Java language which will be talked about later on. Therefore, most of the files in this folder will be JSP (JavaServer Page) which allows the mixed usage of Java and HTML[3].

The second part is a back-end Source Package which uses Java language throughout. The OOP folder consists of JAVA classes that inherit object-oriented characteristics, the detailed explanation of this will be contained in the next part. The Servlet folder contains files that handle function requests from the web page JSP files[4] such as the delete button.



```
@WebServlet(name = "DeleteStatement", urlPatterns = {"/DeleteStatement"})
public class DeleteStatement extends HttpServlet {

    /** Processes requests for both HTTP <code>GET</code> and <code>POST</code> ...9 lines */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {...3 lines }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign o
    /** Handles the HTTP <code>GET</code> method ...8 lines */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {...3 lines }

    /** Handles the HTTP <code>POST</code> method ...8 lines */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
                response.setContentType(string:"text/html;charset=UTF-8");
        String ID = request.getParameter(string:"id");
        try{
            JOconnection.JConnect();

            String sql = "DELETE from financialstatement WHERE id ="+ID;

            response.sendRedirect(string:"index2.jsp");
            JOconnection.stmt.executeUpdate(string:sql);
        }
        catch(Exception e){
            out.println(x: e.toString());
        }
    }
```



```
ComSciIA1
    Web Pages
        META-INF
        WEB-INF
        css
        js
        CompanyView.jsp
        User.jsp
        UserEdit.jsp
        addComReg.jsp
        addCompany.jsp
        addStatement.jsp
        index.html
        index2.jsp
        login.jsp
        signup.html
        signup.jsp
    Source Packages
        OOP
            ComRegDAO.java
            Company.java
            CompanyDAO.java
            CompanyRegistration.java
            DisplayFile.java
            FinancialStatement.java
            JOconnection.java
            StatementDAO.java
            User.java
            UserDAO.java
        Servlet
            DeleteComReg.java
            DeleteCompany.java
            DeleteStatement.java
            UploadComReg.java
            UploadCompany.java
            UploadStatement.java
            UserEdit.java
            getFiles.java
```

[3] "JSP file details - IBM." https://www.ibm.com/docs/en/raa/6.1?topic=pages-jsp-file-details. Accessed 8 Jan. 2024.

[4] "Introduction to Java Servlets - GeeksforGeeks." 12 Sept. 2022, https://www.geeksforgeeks.org/introduction-java-servlets/. Accessed 8 Jan. 2024.

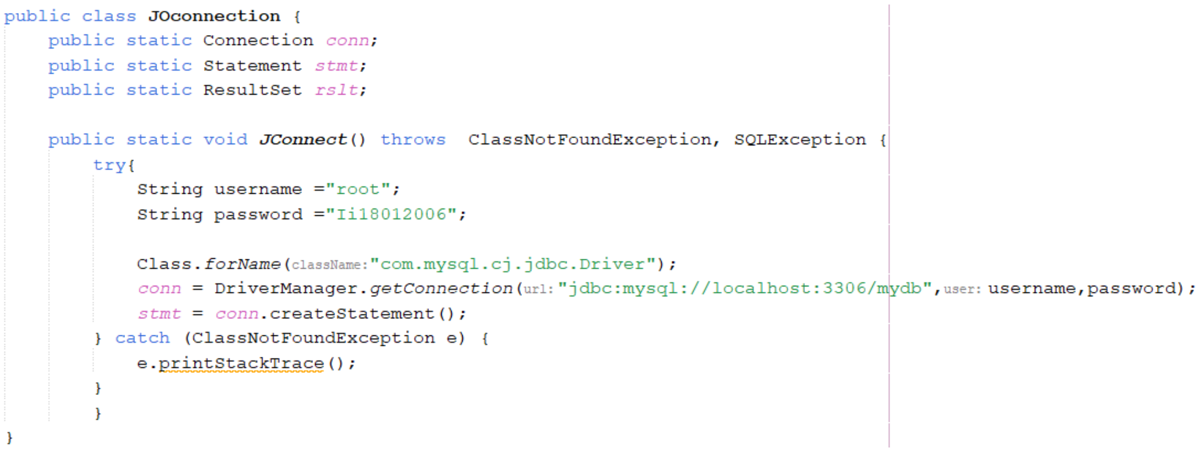
Technique used

1. Database
    - MySQL
        - Connection
        - Table
        - Queries
2. Java Servlet
3. OOP
    - Java
        - Objects
        - Encapsulation
        - Method
4. Dynamic data structure
    - Arraylist
    - Linked list
5. Interface
    - HTML
    - CSS
6. Exception handling
7. Product Extentibility
    - Comment
    - Meaningful varieble

MySQL

MySQL offers a very high-security level for the data stored which fits my client, an accounting company, as the data and information this company contains such as the financial statement is confidential and requires high security. It also has high efficiency even with a lot of queries running or a lot of data stored which fits the project as there will be proportionally more businesses registered with an accounting company than moved to another company; therefore, the amount of data stored will be increasing.

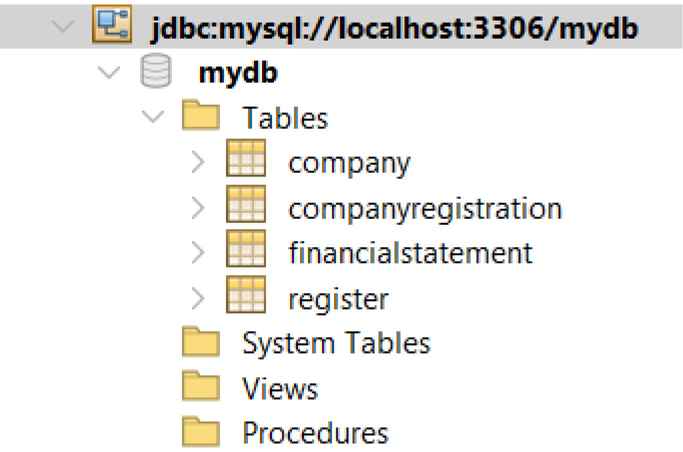
Connection



```
public class JOconnection {
    public static Connection conn;
    public static Statement stmt;
    public static ResultSet rslt;

    public static void JConnect() throws  ClassNotFoundException, SQLException {
        try{
            String username ="root";
            String password ="Ii18012006";

            Class.forName(className:"com.mysql.cj.jdbc.Driver");
            conn = DriverManager.getConnection(url:"jdbc:mysql://localhost:3306/mydb",user: username,password);
            stmt = conn.createStatement();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
        }
}
```

The connection with the database is established using jdbc connector. By making the connection in a class, the connection establish statement won't have to be written again everytime the website need to connect to the database which saves time and resources when coding.

Table

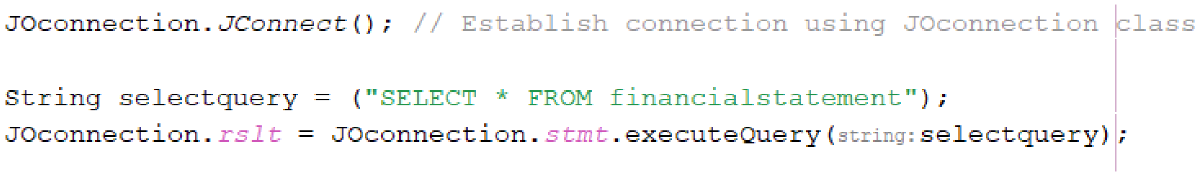


```
jdbc:mysql://localhost:3306/mydb
    mydb
        Tables
            company
            companyregistration
            financialstatement
            register
        System Tables
        Views
        Procedures
```

The table of data includes company, company registration, financial statement, and register which contains company, all company registration, yearly financial statement, and users data consecutively. As company contains aggregation relationship which company registration and financial statement, I added foreign key for company registration table in company column to reference name column in company table and foreign key for financial statement table in company column to reference name column in company table.

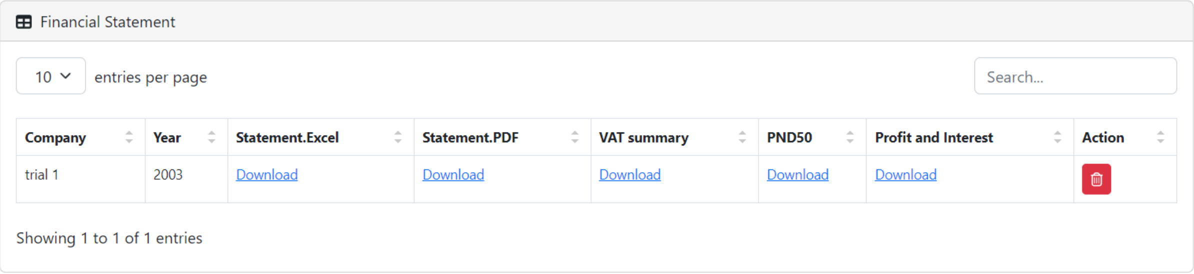
Queries

Select



```
JOconnection.JConnect(); // Establish connection using JOconnection class

String selectquery = ("SELECT * FROM financialstatement");
JOconnection.rslt = JOconnection.stmt.executeQuery(string:selectquery);
```

Select queries are used to retrieve data from the given table in a given column such as in the figure shown selection of all data from the financial statement table. This has been used to show data of yearly financial statement from the database in the form of table which includes all column in the database table by the query "select *" shown below.



Financial Statement

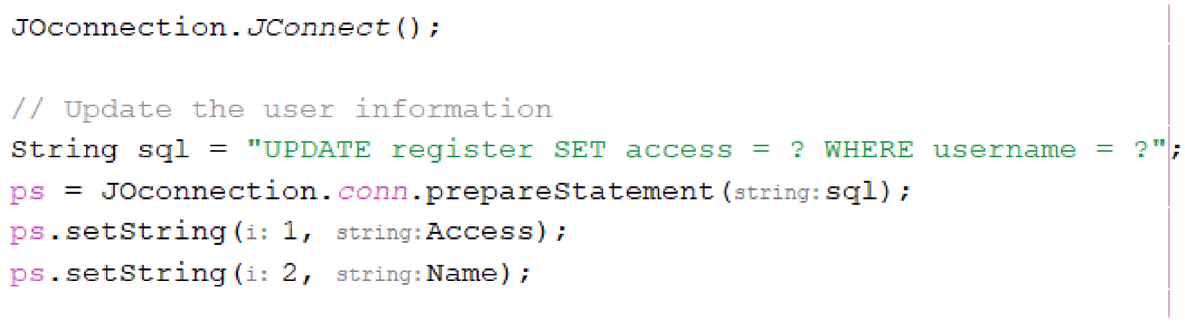
10 entries per page

Search...

Company	Year	Statement.Excel	Statement.PDF	VAT summary	PND50	Profit and Interest	Action
trial 1	2003	Download	Download	Download	Download	Download	🗑

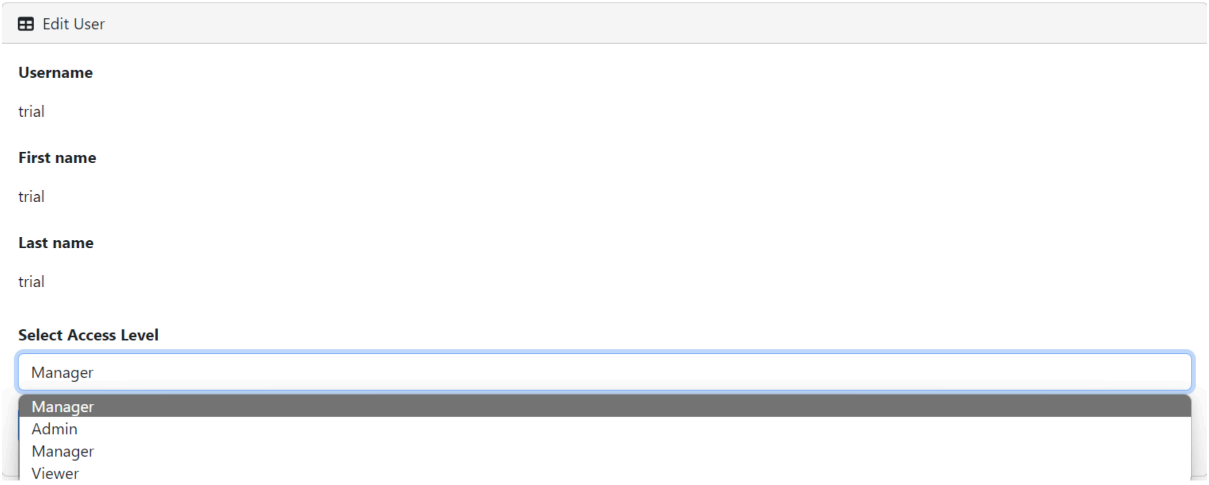
Showing 1 to 1 of 1 entries

Update



```
JOconnection.JConnect();

// Update the user information
String sql = "UPDATE register SET access = ? WHERE username = ?";
ps = JOconnection.conn.prepareStatement(string: sql);
ps.setString(i: 1, string: Access);
ps.setString(i: 2, string: Name);
```

Update queries are used to modify data from the given table in a given column such as in the figure shown the modification of column access in the register table using username as the indication of which row access should be updated. This is implemented into the program in the user management section shown below. After choosing the user role and clicking the submit button, the query will be executed and modify the data in the register database accordingly.



Edit User

Username

trial

First name

trial

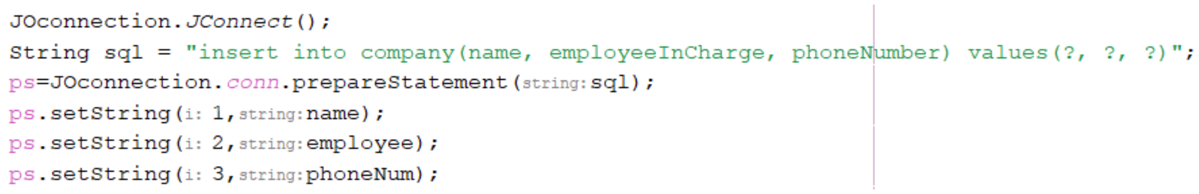
Last name

trial

Select Access Level

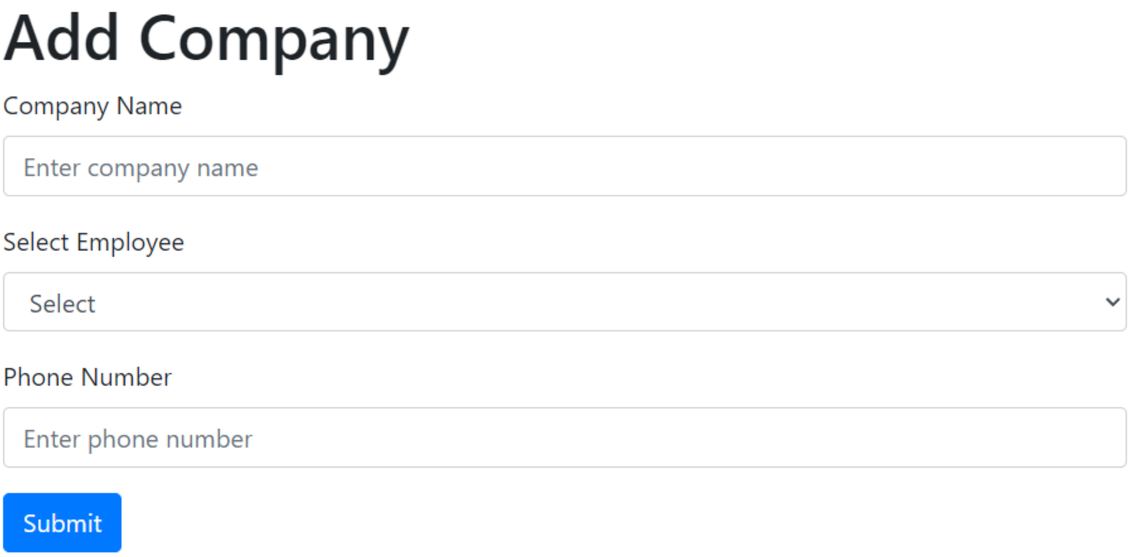
Manager

Insert



```
JOconnection.JConnect();
String sql = "insert into company(name, employeeInCharge, phoneNumber) values(?, ?, ?)";
ps=JOconnection.conn.prepareStatement(string: sql);
ps.setString(i: 1,string: name);
ps.setString(i: 2,string: employee);
ps.setString(i: 3,string: phoneNum);
```

Insert queries are used to add data to the given table such as in the figure shown the addition of name, employee in charge, and phone number data in company table. This is used in the add company function which inputs the requested data for the query shown below.



Add Company

Company Name

Enter company name

Select Employee

Select

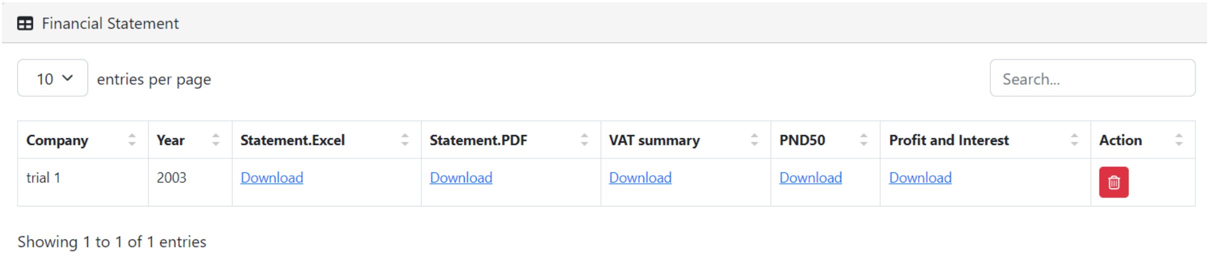
Phone Number

Enter phone number

Delete

```
JOconnection.JConnect();

String sql = "DELETE from financialstatement WHERE id ="+ID;
```

Delete queries is used to delete data from the given table such as in the figure shown the deletion of a financial statement which has an id equal to the input variable. This is used in delete financial statement function shown below; after the delete button in clicked, the financial statement id will be returned and this query will be executed.



Financial Statement

10 entries per page

Company	Year	Statement.Excel	Statement.PDF	VAT summary	PND50	Profit and Interest	Action
trial 1	2003	Download	Download	Download	Download	Download	🗑

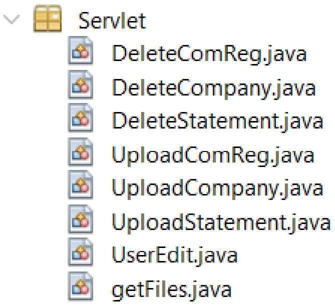
Showing 1 to 1 of 1 entries

Java Servlet

Java Servlet is a Java class which runs on a web server to serve the properties of dynamic websites[5]. It allows the financial company web application to generate pages according to requests from the user such as deleting data from the database and show the newly generated page which shows the modified datafrom the database as shown below[6]:

[5] Jain, Sandeep. "Introduction to Java Servlets." *GeeksforGeeks*, 9 January 2024, https://www.geeksforgeeks.org/introduction-java-servlets/. Accessed 3 March 2024.

[6] "Dynamic Website: How It Works and Should You Build One?" *Cloudinary*, 26 December 2023, https://cloudinary.com/guides/e-commerce-platform/dynamic-website-how-it-works-and-should-you-build-one. Accessed 3 March 2024.

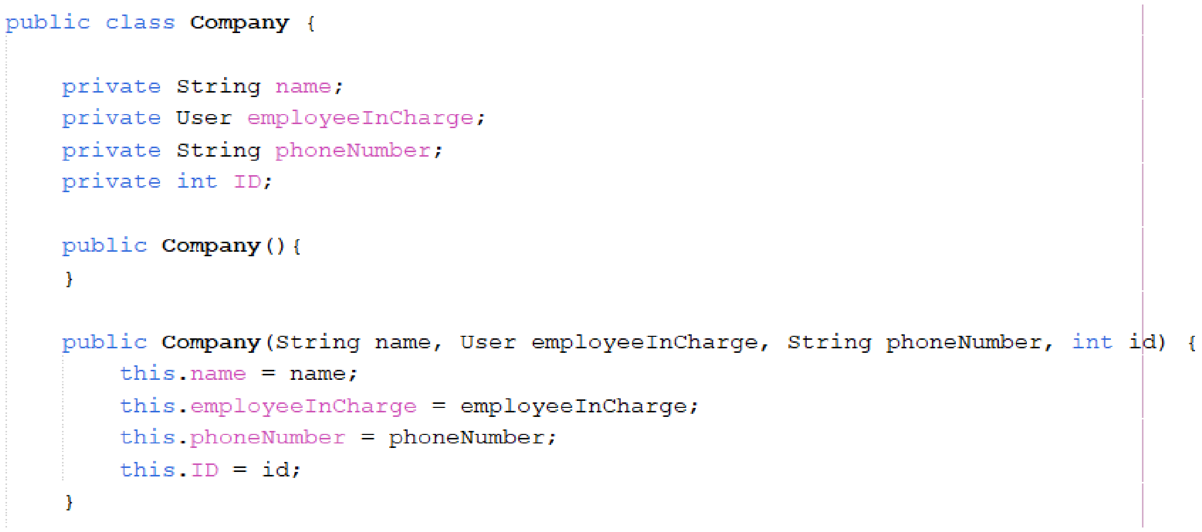


```
Servlet
    DeleteComReg.java
    DeleteCompany.java
    DeleteStatement.java
    UploadComReg.java
    UploadCompany.java
    UploadStatement.java
    UserEdit.java
    getFiles.java
```

<u>OOP - Java</u>

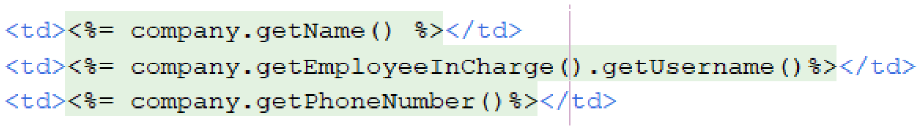
OOP allows data to be represent as object so that different parts of the program can interact and uses method defined throughout without having to access the detail of each other function and variable increasing security, efficiency, and flexibility.

Object



```java
public class Company {

    private String name;
    private User employeeInCharge;
    private String phoneNumber;
    private int ID;

    public Company(){
    }

    public Company(String name, User employeeInCharge, String phoneNumber, int id) {
        this.name = name;
        this.employeeInCharge = employeeInCharge;
        this.phoneNumber = phoneNumber;
        this.ID = id;
    }
```

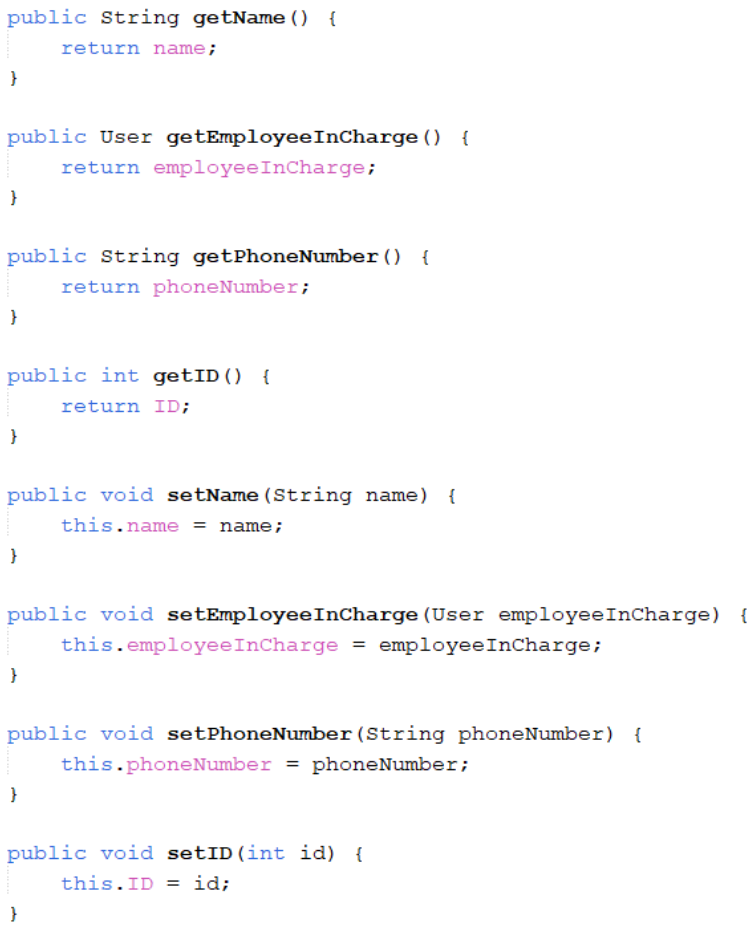
The object Company can be initialized using name, employee in charge, phone number, and id input in other class so that the methods contained in this clas can be used in other classes.



```
<td><%= company.getName() %></td>
<td><%= company.getEmployeeInCharge().getUsername()%></td>
<td><%= company.getPhoneNumber()%></td>
```

In this example, object company is using methos getName(), getEmployeeInCharge(), and getPhoneNumber() which are method that defines in Company class but can be used in other classes.

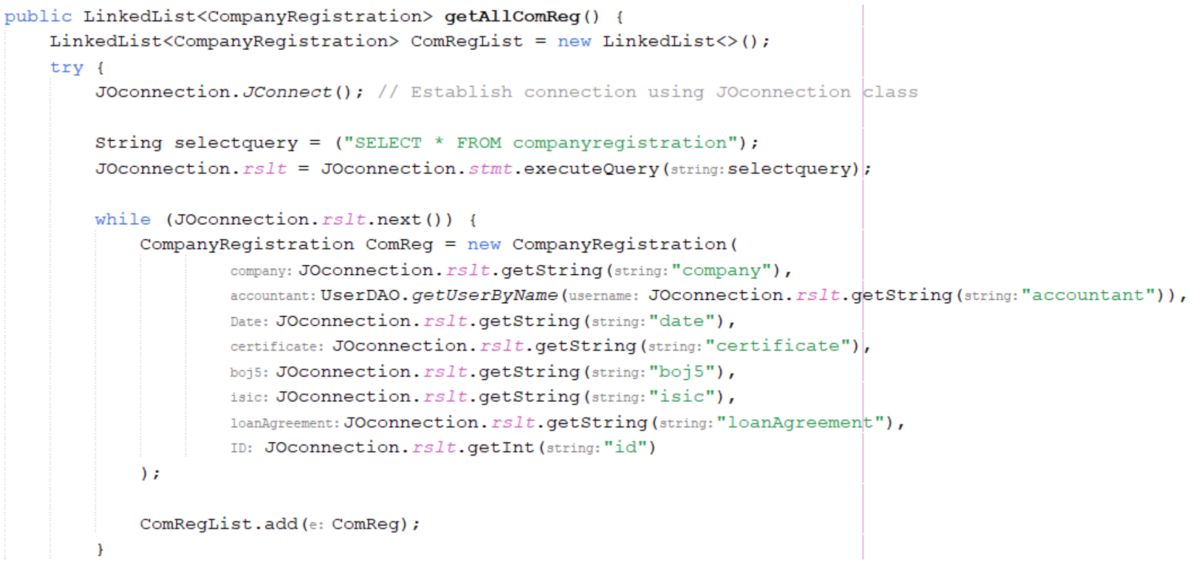
Encapsulation



```
public String getName() {
    return name;
}

public User getEmployeeInCharge() {
    return employeeInCharge;
}

public String getPhoneNumber() {
    return phoneNumber;
}

public int getID() {
    return ID;
}

public void setName(String name) {
    this.name = name;
}

public void setEmployeeInCharge(User employeeInCharge) {
    this.employeeInCharge = employeeInCharge;
}

public void setPhoneNumber(String phoneNumber) {
    this.phoneNumber = phoneNumber;
}

public void setID(int id) {
    this.ID = id;
}
```

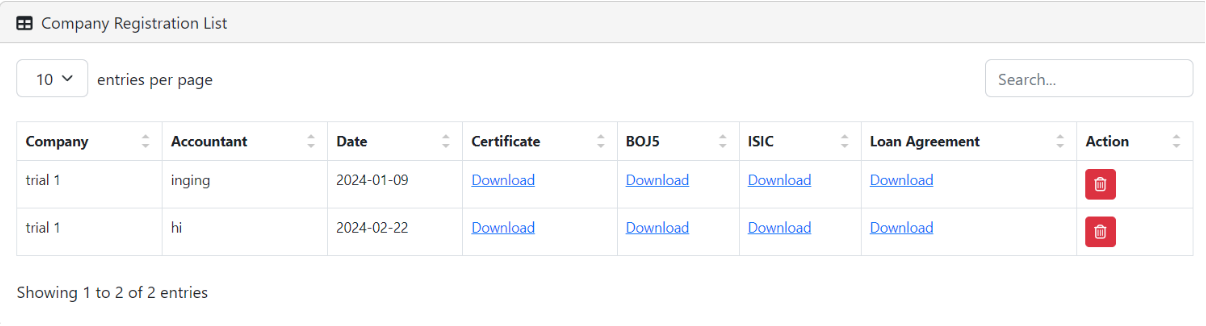
Object Company has private variable of name, employee in charge, phone number, and ID meaning it can't be directly access in other classes but instead can be accessed and modify by method of setters and getters, establishing encapsulation characteristic.

Method



```
public LinkedList<CompanyRegistration> getAllComReg() {
    LinkedList<CompanyRegistration> ComRegList = new LinkedList<>();
    try {
        JOconnection.JConnect(); // Establish connection using JOconnection class

        String selectquery = ("SELECT * FROM companyregistration");
        JOconnection.rslt = JOconnection.stmt.executeQuery(string:selectquery);

        while (JOconnection.rslt.next()) {
            CompanyRegistration ComReg = new CompanyRegistration(
                    company: JOconnection.rslt.getString(string:"company"),
                    accountant: UserDAO.getUserByName(username: JOconnection.rslt.getString(string:"accountant")),
                    Date: JOconnection.rslt.getString(string:"date"),
                    certificate: JOconnection.rslt.getString(string:"certificate"),
                    boj5: JOconnection.rslt.getString(string:"boj5"),
                    isic: JOconnection.rslt.getString(string:"isic"),
                    loanAgreement: JOconnection.rslt.getString(string:"loanAgreement"),
                    ID: JOconnection.rslt.getInt(string:"id")
            );

            ComRegList.add(e: ComReg);
        }
```

Method that has been defined in a class can be access by other class using the characteristic of OOP meaning that the same method and code doesn't need to be rewritten and repeat every time a method is needed, reduce redundancy and memory space for the code. The provided method getAllComReg() is used to retrieve all data in the company registration table in the database to use when representing data in the interface shown below.



Company Registration List

10 entries per page

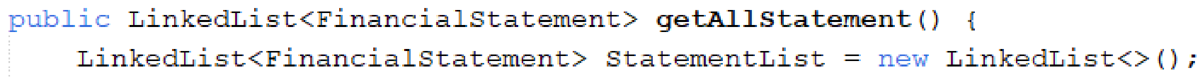
Search...

Company	Accountant	Date	Certificate	BOJ5	ISIC	Loan Agreement	Action
trial 1	inging	2024-01-09	Download	Download	Download	Download	🗑
trial 1	hi	2024-02-22	Download	Download	Download	Download	🗑

Showing 1 to 2 of 2 entries

Dynamic data structure

Linked List

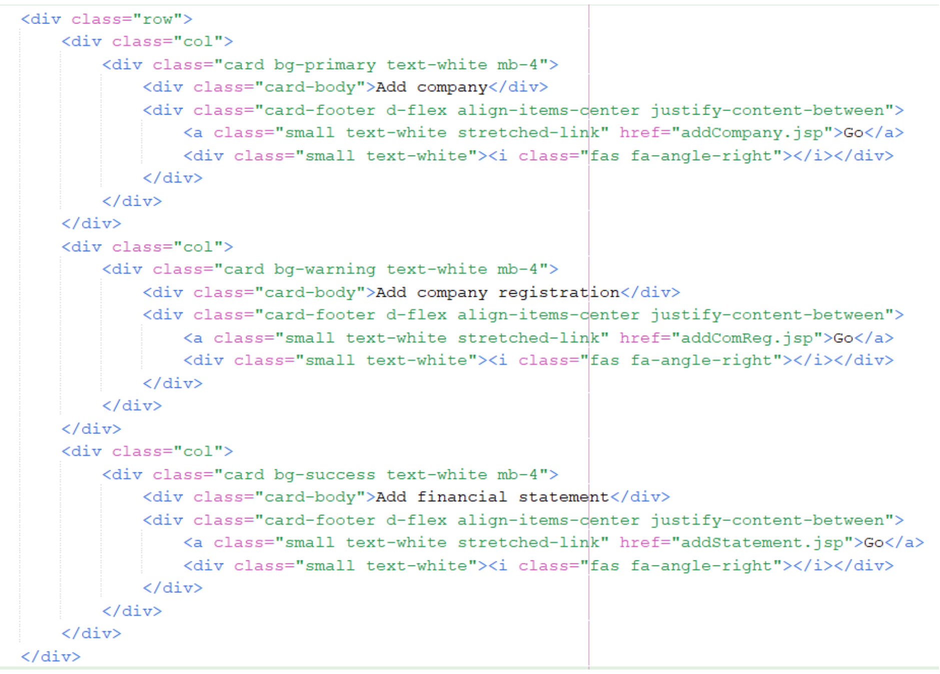


```
public LinkedList<FinancialStatement> getAllStatement() {
    LinkedList<FinancialStatement> StatementList = new LinkedList<>();
```

It is used to stored data represented in a row with each node pointing to the next fitting for financial statement representation as the financial statement is produced continuously once a year as well we the linked list being dynamic so it doen’t limit amount of financial statement stored.

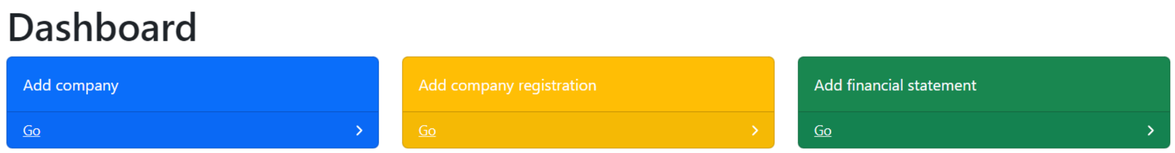
Interface

HTML



```
<div class="row">
    <div class="col">
        <div class="card bg-primary text-white mb-4">
            <div class="card-body">Add company</div>
            <div class="card-footer d-flex align-items-center justify-content-between">
                <a class="small text-white stretched-link" href="addCompany.jsp">Go</a>
                <div class="small text-white"><i class="fas fa-angle-right"></i></div>
            </div>
        </div>
    </div>
    <div class="col">
        <div class="card bg-warning text-white mb-4">
            <div class="card-body">Add company registration</div>
            <div class="card-footer d-flex align-items-center justify-content-between">
                <a class="small text-white stretched-link" href="addComReg.jsp">Go</a>
                <div class="small text-white"><i class="fas fa-angle-right"></i></div>
            </div>
        </div>
    </div>
    <div class="col">
        <div class="card bg-success text-white mb-4">
            <div class="card-body">Add financial statement</div>
            <div class="card-footer d-flex align-items-center justify-content-between">
                <a class="small text-white stretched-link" href="addStatement.jsp">Go</a>
                <div class="small text-white"><i class="fas fa-angle-right"></i></div>
            </div>
        </div>
    </div>
</div>
```

HTML is used to build structure and display information in a website. It can also be integrated with other languages such as javascript or CSS which can be used for aesthetic purposes. The code above represents HTML code used to build buttons for adding a new company, company registration, and financial statement shown below.



Dashboard

Add company

Go ›

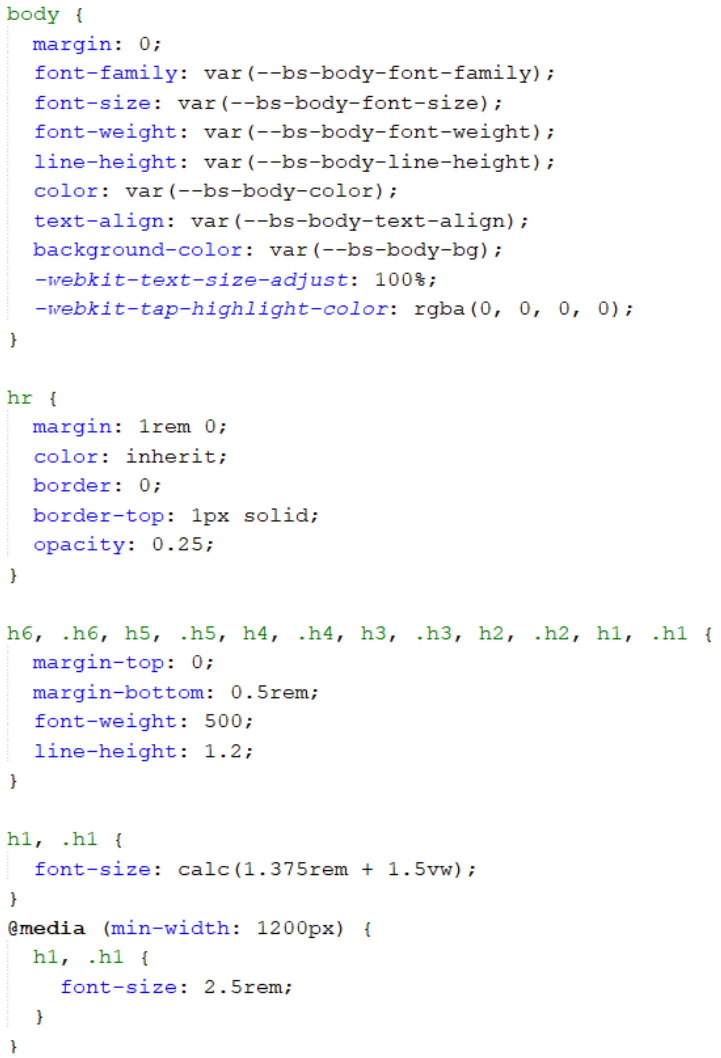
Add company registration

Go ›

Add financial statement

Go ›

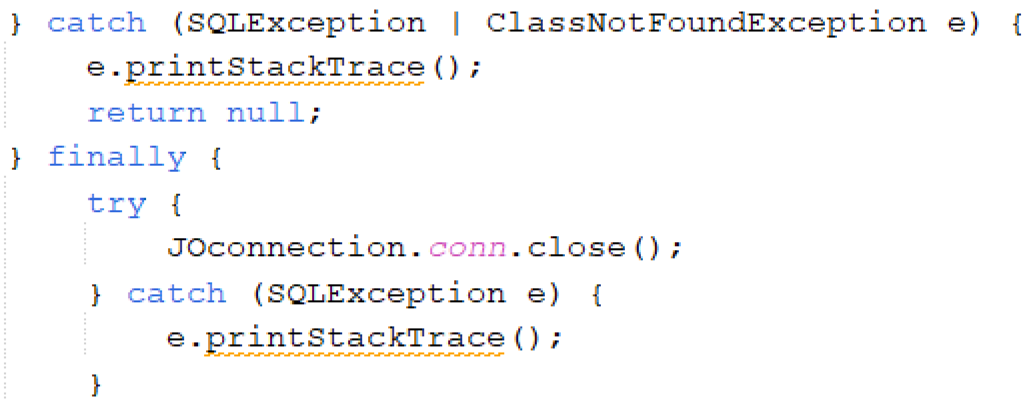
CSS



```
body {
  margin: 0;
  font-family: var(--bs-body-font-family);
  font-size: var(--bs-body-font-size);
  font-weight: var(--bs-body-font-weight);
  line-height: var(--bs-body-line-height);
  color: var(--bs-body-color);
  text-align: var(--bs-body-text-align);
  background-color: var(--bs-body-bg);
  -webkit-text-size-adjust: 100%;
  -webkit-tap-highlight-color: rgba(0, 0, 0, 0);
}

hr {
  margin: 1rem 0;
  color: inherit;
  border: 0;
  border-top: 1px solid;
  opacity: 0.25;
}

h6, .h6, h5, .h5, h4, .h4, h3, .h3, h2, .h2, h1, .h1 {
  margin-top: 0;
  margin-bottom: 0.5rem;
  font-weight: 500;
  line-height: 1.2;
}

h1, .h1 {
  font-size: calc(1.375rem + 1.5vw);
}
@media (min-width: 1200px) {
  h1, .h1 {
    font-size: 2.5rem;
  }
}
```

CSS code is used for styling the website to makes it more appealing to use. The code above is fraction of CSS Boostrap code which is a free front-end framework. It helps improve efficiency and save time to develop the website as the code are provided and ready to be implemented into the website.

Exception Handling

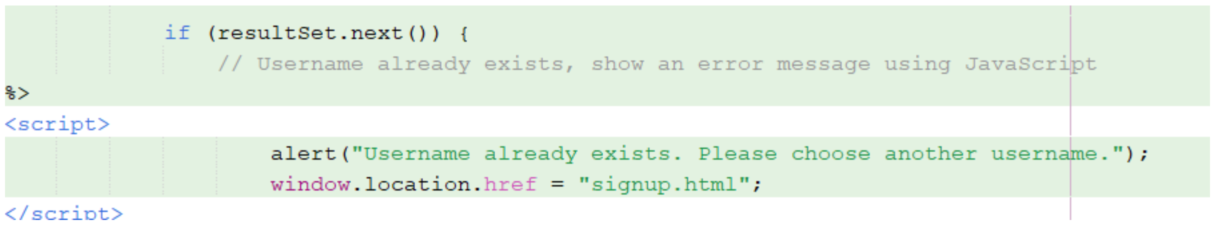


```
} catch (SQLException | ClassNotFoundException e) {
    e.printStackTrace();
    return null;
} finally {
    try {
        JOconnection.conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
```

The try-catch code is used to handle exceptions in the program as shown in the figure above. It helps maintain normal flow of the program and helps identify errors in the program so it can be fixed accordingly and improve efficiency of developing the program.

Product extentibility

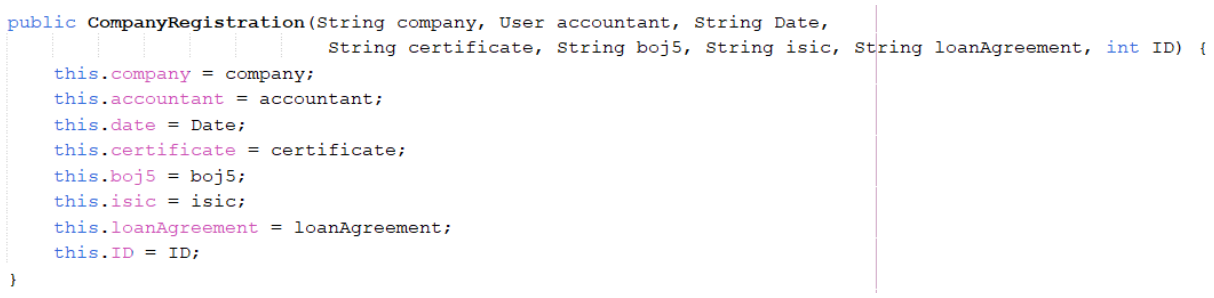
Comment



```
            if (resultSet.next()) {
                // Username already exists, show an error message using JavaScript
%>
<script>
                    alert("Username already exists. Please choose another username.");
                    window.location.href = "signup.html";
</script>
```

Commenting helps other developer that wants to further develop the code after understand each main function and code on what each code does. For example, there is a comment to indicate that the following code will be javascript code used for sending alert massage when the input username already exist in the database.

Meaningful variable



```
public CompanyRegistration(String company, User accountant, String Date,
                           String certificate, String boj5, String isic, String loanAgreement, int ID) {
    this.company = company;
    this.accountant = accountant;
    this.date = Date;
    this.certificate = certificate;
    this.boj5 = boj5;
    this.isic = isic;
    this.loanAgreement = loanAgreement;
    this.ID = ID;
}
```

Using meaningful variables are also important in helping other developer understand the code as other developers will instantly know what each variable is used for. For example, variable company, accountant, date, certificate, etc. is used in company registration because these are the essential component contained in a company registration.

Word count: 1223 words

Criteria E: Evaluation

<table>
<tr><th>Success Criteria</th><th>Evaluation and outcome</th></tr>
<tr><td>

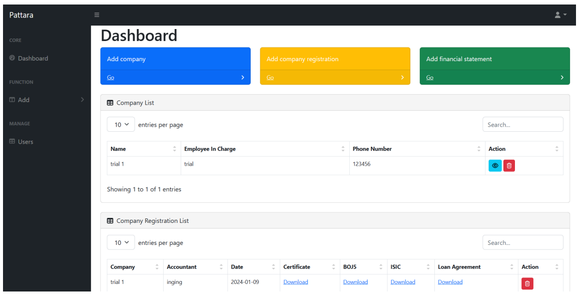
Login page for different types of access

- Administrator - can edit, delete, and access everything
- Manager - can add and access but not delete
- Employees - can only access

</td><td>

Successful implementation of the function helps improve the security level of data access and modification according to the client.

Administrator interface:



Dashboard

Company List

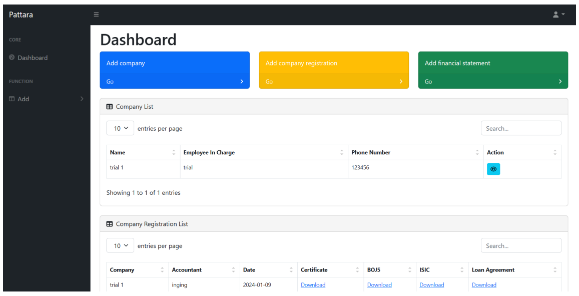
<table>
<tr><th>Name</th><th>Employee In Charge</th><th>Phone Number</th><th>Action</th></tr>
<tr><td>trial 1</td><td>trial</td><td>123456</td><td>👁 🗑</td></tr>
</table>

Showing 1 to 1 of 1 entries

Company Registration List

<table>
<tr><th>Company</th><th>Accountant</th><th>Date</th><th>Certificate</th><th>BOJ5</th><th>ISIC</th><th>Loan Agreement</th><th>Action</th></tr>
<tr><td>trial 1</td><td>inging</td><td>2024-01-09</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td><td>🗑</td></tr>
</table>

Manager interface:



Dashboard

Company List

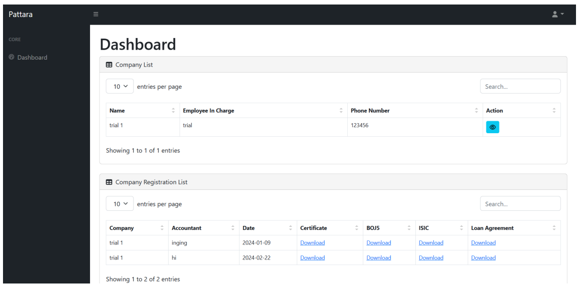
<table>
<tr><th>Name</th><th>Employee In Charge</th><th>Phone Number</th><th>Action</th></tr>
<tr><td>trial 1</td><td>trial</td><td>123456</td><td>👁</td></tr>
</table>

Showing 1 to 1 of 1 entries

Company Registration List

<table>
<tr><th>Company</th><th>Accountant</th><th>Date</th><th>Certificate</th><th>BOJ5</th><th>ISIC</th><th>Loan Agreement</th></tr>
<tr><td>trial 1</td><td>inging</td><td>2024-01-09</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td></tr>
</table>

Employees interface:



Dashboard

Company List

<table>
<tr><th>Name</th><th>Employee In Charge</th><th>Phone Number</th><th>Action</th></tr>
<tr><td>trial 1</td><td>trial</td><td>123456</td><td>👁</td></tr>
</table>

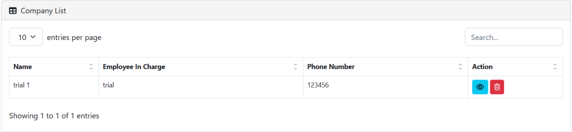
Showing 1 to 1 of 1 entries

Company Registration List

<table>
<tr><th>Company</th><th>Accountant</th><th>Date</th><th>Certificate</th><th>BOJ5</th><th>ISIC</th><th>Loan Agreement</th></tr>
<tr><td>trial 1</td><td>inging</td><td>2024-01-09</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td></tr>
<tr><td>trial 1</td><td>hi</td><td>2024-02-22</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td></tr>
</table>

Showing 1 to 2 of 2 entries

</td></tr>
<tr><td>Company list interface</td><td>Successful implementation of the function helps display the fundamental function of the software and improve the organization level of data displays according to the client.</td></tr>
</table>

<table>
<tr>
<td></td>
<td>


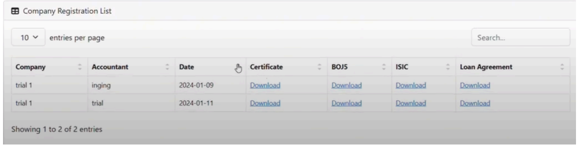
Company List

10 entries per page

<table>
<tr><th>Name</th><th>Employee In Charge</th><th>Phone Number</th><th>Action</th></tr>
<tr><td>trial 1</td><td>trial</td><td>123456</td><td>👁 🗑</td></tr>
</table>

Showing 1 to 1 of 1 entries
</td>
</tr>
<tr>
<td>Sort all company registrations by date</td>
<td>
Successful implementation of the function helps categorize and improve accessibility to the company registration which will be used sorted by the date registered according to the client.

Sort by oldest to newest:



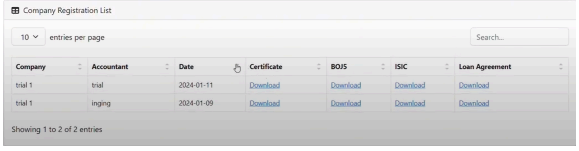
Company Registration List

10 entries per page

<table>
<tr><th>Company</th><th>Accountant</th><th>Date</th><th>Certificate</th><th>BOJ5</th><th>ISIC</th><th>Loan Agreement</th></tr>
<tr><td>trial 1</td><td>inging</td><td>2024-01-09</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td></tr>
<tr><td>trial 1</td><td>trial</td><td>2024-01-11</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td></tr>
</table>

Showing 1 to 2 of 2 entries

Sort by newest to oldest:

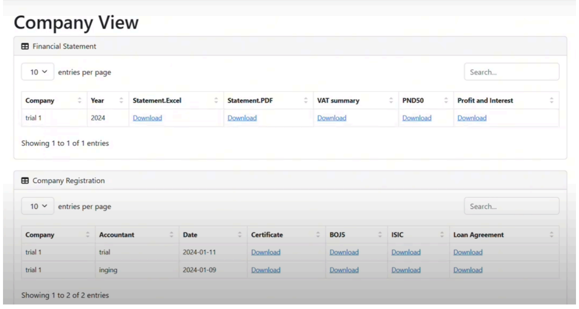


Company Registration List

10 entries per page

<table>
<tr><th>Company</th><th>Accountant</th><th>Date</th><th>Certificate</th><th>BOJ5</th><th>ISIC</th><th>Loan Agreement</th></tr>
<tr><td>trial 1</td><td>trial</td><td>2024-01-11</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td></tr>
<tr><td>trial 1</td><td>inging</td><td>2024-01-09</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td></tr>
</table>

Showing 1 to 2 of 2 entries
</td>
</tr>
<tr>
<td>Past company registration list for each company</td>
<td>
Successful implementation of the function helps display the fundamental function of the software and improve the organization level of data displays according to the client.



Company View

Financial Statement

10 entries per page

<table>
<tr><th>Company</th><th>Year</th><th>Statement.Excel</th><th>Statement.PDF</th><th>VAT summary</th><th>PND50</th><th>Profit and Interest</th></tr>
<tr><td>trial 1</td><td>2024</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td></tr>
</table>

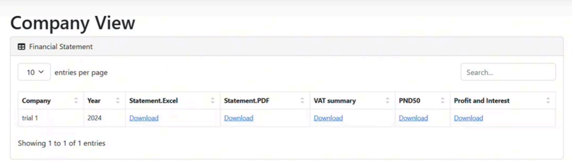
Showing 1 to 1 of 1 entries

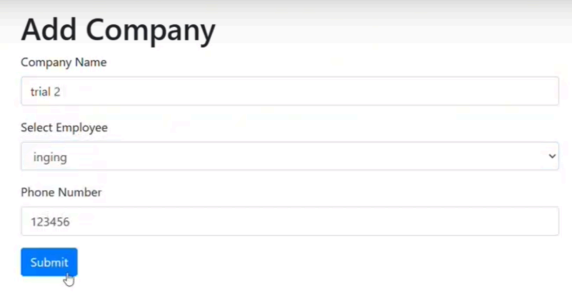
Company Registration

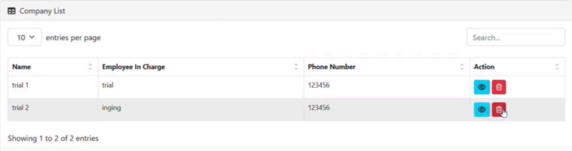
10 entries per page

<table>
<tr><th>Company</th><th>Accountant</th><th>Date</th><th>Certificate</th><th>BOJ5</th><th>ISIC</th><th>Loan Agreement</th></tr>
<tr><td>trial 1</td><td>trial</td><td>2024-01-11</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td></tr>
<tr><td>trial 1</td><td>inging</td><td>2024-01-09</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td></tr>
</table>

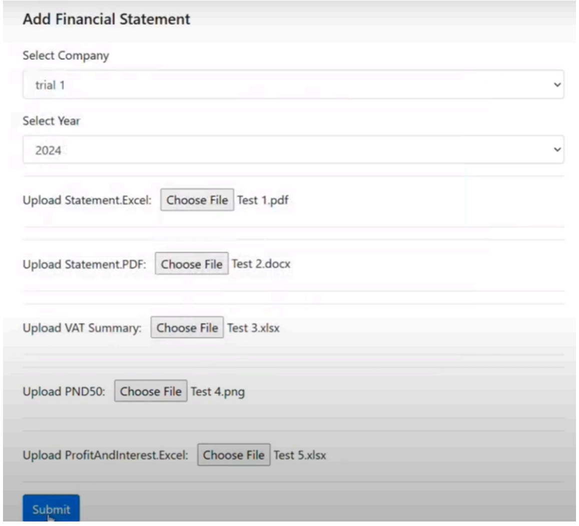
Showing 1 to 2 of 2 entries
</td>
</tr>
<tr>
<td>The financial statement list for each company</td>
<td>Successful implementation of the function helps display the fundamental function of the software and improve the organization</td>
</tr>
</table>

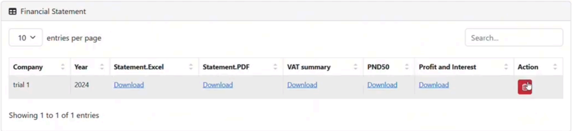
<table>
<tr>
<td></td>
<td>level of data displays according to the client.

Company View
Financial Statement
10 entries per page
Search...
<table>
<tr><th>Company</th><th>Year</th><th>Statement.Excel</th><th>Statement.PDF</th><th>VAT summary</th><th>PND50</th><th>Profit and Interest</th></tr>
<tr><td>trial 1</td><td>2024</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td></tr>
</table>
Showing 1 to 1 of 1 entries</td>
</tr>
<tr>
<td>Add and remove a company from the list</td>
<td>Successful implementation of the function helps display the fundamental function of the software requested by the client.

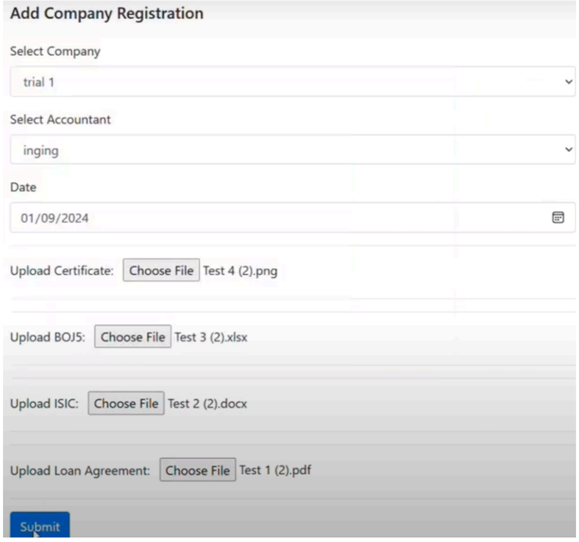
Add company function:

Add Company
Company Name: trial 2
Select Employee: inging
Phone Number: 123456

Delete company function:

Company List
10 entries per page
Search...
<table>
<tr><th>Name</th><th>Employee In Charge</th><th>Phone Number</th><th>Action</th></tr>
<tr><td>trial 1</td><td>trial</td><td>123456</td><td>👁 🗑</td></tr>
<tr><td>trial 2</td><td>inging</td><td>123456</td><td>👁 🗑</td></tr>
</table>
Showing 1 to 2 of 2 entries</td>
</tr>
<tr>
<td>Add and remove a financial statement from the list</td>
<td>Successful implementation of the function helps display the fundamental function of the software requested by the client.

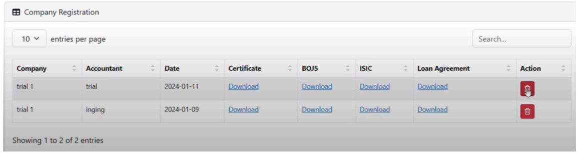
Add financial statement function:</td>
</tr>
</table>

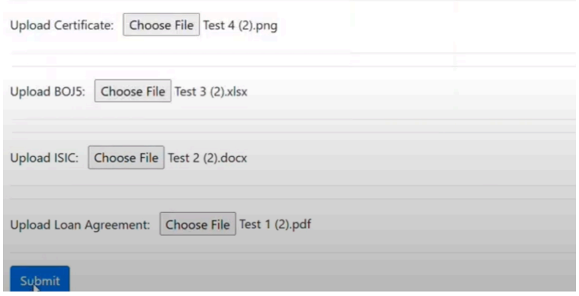
<table>
<tr>
<td></td>
<td>

Add Financial Statement
Select Company: trial 1
Select Year: 2024
Upload Statement.Excel: Choose File Test 1.pdf
Upload Statement.PDF: Choose File Test 2.docx
Upload VAT Summary: Choose File Test 3.xlsx
Upload PND50: Choose File Test 4.png
Upload ProfitAndInterest.Excel: Choose File Test 5.xlsx
Submit

Delete financial statement function:

Financial Statement
10 entries per page
Search...
<table>
<tr><th>Company</th><th>Year</th><th>Statement.Excel</th><th>Statement.PDF</th><th>VAT summary</th><th>PND50</th><th>Profit and Interest</th><th>Action</th></tr>
<tr><td>trial 1</td><td>2024</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td><td></td></tr>
</table>
Showing 1 to 1 of 1 entries
</td>
</tr>
<tr>
<td>Add and remove company registration from the list</td>
<td>
Successful implementation of the function helps display the fundamental function of the software requested by the client.

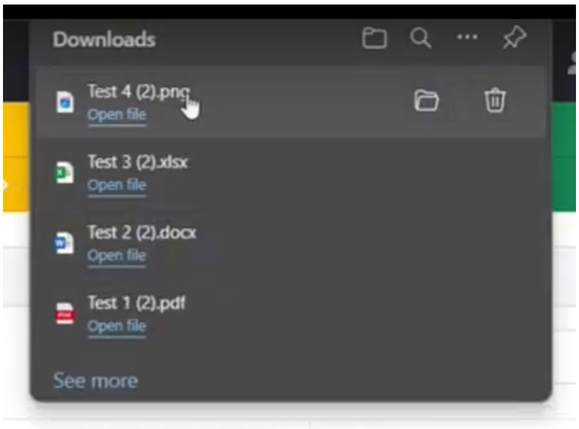
Add company registration function:

Add Company Registration
Select Company: trial 1
Select Accountant: inging
Date: 01/09/2024
Upload Certificate: Choose File Test 4 (2).png
Upload BOJ5: Choose File Test 3 (2).xlsx
Upload ISIC: Choose File Test 2 (2).docx
Upload Loan Agreement: Choose File Test 1 (2).pdf
Submit

Delete company registration function:
</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>

Company Registration
10 entries per page
Search...
<table>
<tr><th>Company</th><th>Accountant</th><th>Date</th><th>Certificate</th><th>BOJ5</th><th>ISIC</th><th>Loan Agreement</th><th>Action</th></tr>
<tr><td>trial 1</td><td>trial</td><td>2024-01-11</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td><td></td></tr>
<tr><td>trial 1</td><td>inging</td><td>2024-01-09</td><td>Download</td><td>Download</td><td>Download</td><td>Download</td><td></td></tr>
</table>
Showing 1 to 2 of 2 entries
</td>
</tr>
<tr>
<td>Be able to upload different types of files and store it</td>
<td>Successful implementation of the function as the files used by this accounting company are in different formats such as PDF or Excel requested by the client.

Upload documents for png, .xlsx, .docx, and PDF files format:

Upload Certificate: Choose File Test 4 (2).png
Upload BOJ5: Choose File Test 3 (2).xlsx
Upload ISIC: Choose File Test 2 (2).docx
Upload Loan Agreement: Choose File Test 1 (2).pdf

</td>
</tr>
<tr>
<td>Be able to download the stored different types of files</td>
<td>Successful implementation of the function as the files used by this accounting company are in different formats such as PDF or Excel requested by the client.

Download documents for png, .xlsx, .docx, and PDF files format:


Test 4 (2).png

Test 3 (2).xlsx

Test 2 (2).docx

Test 1 (2).pdf

</td>
</tr>
</table>

Recommendation for improvements

According to the final interview with the client, references interview 3 in the appendix, further improvements could be made by adding an edit function for company details. This allows more flexibility in the program as many companies change their details such as name and needs to be updated in the program as well. By implementing this functionality, it reduces administrative work and ensures that the program is always up-to-date.

Another function that could be implemented is a tax calculation function which helps calculate basic income and business tax for each company based on their income and revenue data inputted into the program. This would lower the margin for human errors in the calculation and help increase the efficiency of the client's company by reducing the time taken to calculate these taxes.

An additional useful function for improvement is adding a confirmation window or pop-up when users click on the delete function to prevent unintentional data loss which will reduce the possibility of accidental deletions by asking users to confirm their action before it is carried out as it is irreversible. This feature increases software usability and develops confidence among users and also exhibits a user-centric design to enhance user experience and boost confidence in the reliability of the program.

Word count: 212 words

Appendix

Interview 1

Candidate: Please introduce yourself and your job.

Client: My name is Pattaraporn and I am the owner of an accounting firm

Candidate: Can you describe your work?

Client: Yes sure, my accounting firm is handling all accounting aspects of our client's business from producing financial statements to calculating and submitting their taxes.

Candidate: Are there any problems you face within your working environment that you would like to introduce software into use to fix the problem?

Client: Yes, my company is currently using paper-based accounting which stores files in physical paper and sometimes it is hard to access the old data as we have to go through all the files we have from many businesses in many years.

Client: Another problem is about our clients' company registration which may change in some years and affect the information we need to fill out when submitting their taxes. Currently, we need to go back and check the company registration for each of our client's businesses, whether it is changed or not, as my employee who handles the change in company registration and the one who submitted taxes are in different teams so I would like to have a program which put the newly changed company registration in a list.

Candidate: When you said 'newly changed company registration' do you mean change within a year or what time period maybe I can make it reset every new year?

Client: No because each company will have different deadlines for closing their financial statements throughout the year so it will be better if we are able to reset the list ourselves.

Candidate: Got it, so I will add a reset button to reset the newly changed company registration. From what I discussed with you so far, you want a database to store all the files from different companies in different years as well as a section for company registration. Is there anything else you would like to add?

Client: No, that's all I need

Candidate: Okay. That is all the questions I have for you. Thank you so much for your time.

Interview 2

Candidate: Here is a rough design of what the website interface will looks like, What do you think about it.

Client: That looks nice, I like it, but since our first interaction I was thinking about how the company registration is sorted. Is it possible to change the company registration sorting to just be sorted by the date from newest to oldest and not having reset button anymore.

Candidate: Yes, I can do that for you. Anything else

Client: No, that’s it.

Interview 3

Candidate: Here is the final website, what do you think about it.

Client: Oh! It looks nice. I like how everyone can create their own account and admin can just manage their access so it is less time consuming rather than having admin creating account for everyone individually.

Candidate: Are there any changes you wanna make?

Client: This program looks good for basic documentations but you could improve it to be able to edit the company details as well because some of the businesses do change their names occasionally.

Candidate: Got it, thank you so much.

Library used

- https://dev.mysql.com/downloads/connector/j/

Works Cited

Home, https://github.com/Java-Hub18/Servlet-JSP/tree/11ba5e0f8d56be4e8a847ec3d4df1f99623ae2c0/ServletUploadDownloadFile. Accessed 8 January 2024.

Arroyo, Martin. ",." *, - YouTube*, 22 September 2021, https://www.youtube.com/watch?v=dIFIKjozsSc&ab_channel=CSITBAMU. Accessed 8 January 2024.

Arroyo, Martin. ",." *, - YouTube*, 22 September 2021, https://github.com/y2kstack/Archives/blob/b8fe4d0b87efd27199828ffadb8d2b2622100ca5/_posts/2019-01-26-Project-part1-signuppage.md. Accessed 8 January 2024.

Arroyo, Martin. ",." *, - YouTube*, 22 September 2021, https://github.com/y2kstack/Archives/blob/b8fe4d0b87efd27199828ffadb8d2b2622100ca5/_posts/2019-01-27-Project-part2-login-page.md. Accessed 8 January 2024.

"Bootstrap html snippet. light user list." *Bootdey*, 21 October 2020, https://www.bootdey.com/snippets/view/light-user-list. Accessed 8 January 2024.

"ChatGPT", 13 Feb. version, OpenAI, 5 January. 2024, chat.openai.com.

"Forms · Bootstrap." *Bootstrap*, https://getbootstrap.com/docs/4.0/components/forms/. Accessed 8 January 2024.

"Live Demo: Bootstrap User Management Data Table." *Tutorial Republic*, https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=user-management-data-table. Accessed 8 January 2024.

"SB Admin - Free Bootstrap Admin Template." *Start Bootstrap*, https://startbootstrap.com/template/sb-admin. Accessed 8 January 2024.

Code

```
<%@page import="OOP.FinancialStatement"%>
<%@page import="OOP.StatementDAO"%>
<%@page import="OOP.CompanyDAO"%>
<%@page import="OOP.Company"%>
<%@page import="OOP.User"%>
<%@page import="OOP.ComRegDAO"%>
<%@page import="OOP.CompanyRegistration"%>
<%@page import="java.util.LinkedList" %>
<!DOCTYPE html>
<!-- Dashboard template from https://startbootstrap.com/template/sb-admin -->
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
/>
        <meta name="description" content="" />
        <meta name="author" content="" />
        <title>Dashboard</title>
        <link href="https://cdn.jsdelivr.net/npm/simple-datatables@7.1.2/dist/style.min.css"
rel="stylesheet" />
        <link href="css/bootstrap.css" rel="stylesheet" />
        <link href="css/bootstrap-grid.css" rel="stylesheet" />
        <link href="css/bootstrap-reboot.css" rel="stylesheet" />
        <link href="css/Home.css" rel="stylesheet" />
        <script src="https://use.fontawesome.com/releases/v6.3.0/js/all.js"
crossorigin="anonymous"></script>
```

```
  </head>

  <body class="sb-nav-fixed">

    <nav class="sb-topnav navbar navbar-expand navbar-dark bg-dark">

      <!-- Navbar Brand-->

      <a class="navbar-brand ps-3" href="index2.html">Pattara</a>


      <!-- Sidebar Toggle-->

      <button class="btn btn-link btn-sm order-1 order-lg-0 me-4 me-lg-0"
id="sidebarToggle"><i class="fas fa-bars"></i></button>

      <!-- Navbar-->

      <ul class="navbar-nav ms-auto me-4">

        <li class="nav-item dropdown">

          <a class="nav-link dropdown-toggle" id="navbarDropdown" href="#"
role="button" data-bs-toggle="dropdown" aria-expanded="false"><i class="fas fa-user
fa-fw"></i></a>

          <ul class="dropdown-menu dropdown-menu-end"
aria-labelledby="navbarDropdown">

            <li><a class="dropdown-item" href="index.html">Logout</a></li>

          </ul>

        </li>

      </ul>

    </nav>

    <div id="layoutSidenav">

      <div id="layoutSidenav_nav">

        <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">

          <div class="sb-sidenav-menu">

            <div class="nav">

              <div class="sb-sidenav-menu-heading">Core</div>
```

```
                            <a class="nav-link" href="index2.jsp">
                                <div class="sb-nav-link-icon"><i class="fas fa-tachometer-alt"></i></div>
                                Dashboard
                            </a>

                            <%
                                User login = (User) session.getAttribute("login");
                                if (login.isManager() || login.isAdmin()){
                            %>
                            <div class="sb-sidenav-menu-heading">Function</div>
                            <a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">
                                <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
                                Add
                                <div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>
                            </a>
                            <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne" data-bs-parent="#sidenavAccordion">
                                <nav class="sb-sidenav-menu-nested nav">
                                    <a class="nav-link" href="addCompany.jsp">Add company</a>
                                    <a class="nav-link" href="addComReg.jsp">Add company registration</a>
                                    <a class="nav-link" href="addStatement.jsp">Add financial statement</a>
                                </nav>
                            </div>
                            <%
```

```
                }
                if (login.isAdmin()){
            %>
            <div class="sb-sidenav-menu-heading">Manage</div>
            <a class="nav-link" href="User.jsp">
                <div class="sb-nav-link-icon"><i class="fas fa-table"></i></div>
                Users
            </a>
            <%
                }
            %>
        </div>
    </div>
</nav>
</div>
<div id="layoutSidenav_content">

    <main>
        <div class="container-fluid px-4">
            <h1 class="mt-4">Company View</h1>
            <div class="card mb-4">
                <div class="card-header">
                    <i class="fas fa-table me-1"></i>
                    Financial Statement
                </div>
                <div class="card-body">
                    <table id="datatablesSimple">
                        <thead>
                    <tr>
                      <th>Company</th>
                      <th>Year</th>
                      <th>Statement.Excel</th>
                      <th>Statement.PDF</th>
                      <th>VAT summary</th>
                      <th>PND50</th>
                      <th>Profit and Interest</th>
                        <%
                          if (login.isAdmin()){
                        %>
                      <th>Action</th>
                        <%
                          }
                        %>
                    </tr>
                  </thead>
                  <tbody>
                    <%
                      StatementDAO statementDAO = new StatementDAO();
                      LinkedList<FinancialStatement> statementList =
statementDAO.getAllStatement();
                      for(FinancialStatement statement : statementList) {

if(statement.getCompany().equals(CompanyDAO.getCompanyByID(Integer.parseInt(request
.getParameter("id"))).getName())) {
                    %>
                    <tr>
```

```
                        <td><%= statement.getCompany() %></td>
                        <td><%= statement.getYear()%></td>
                        <td><a href="getFiles?fileName=<%=
statement.getExcelStatement()%>">Download</a></td>
                        <td><a href="getFiles?fileName=<%=
statement.getPdfStatement()%>">Download</a></td>
                        <td><a href="getFiles?fileName=<%=
statement.getVatSummary()%>">Download</a></td>
                        <td><a href="getFiles?fileName=<%=
statement.getPnd50()%>">Download</a></td>
                        <td><a href="getFiles?fileName=<%=
statement.getExcelProfitAndInterest()%>">Download</a></td>
                        <%
                            if (login.isAdmin()){
                        %>
                        <td class="candidate-list-favourite-time text-right">
                            <form action="DeleteStatement">
                                <button class="btn btn-danger btn-sm" data-toggle="tooltip"
title="Delete"
                                        name="id" value="<%= statement.getID() %>">
                                    <i class="far fa-trash-alt"></i>
                                </button>
                            </form>
                        </td>
                        <%
                            }

                        %>
```

```
                </tr>
                <%
                        }
                    }
                %>
            </tbody>
        </table>
    </div>
</div>

<div class="card mb-4">
    <div class="card-header">
        <i class="fas fa-table me-1"></i>
        Company Registration
    </div>
    <div class="card-body">
        <table id="datatablesSimple1">
            <thead>
                <tr>
                    <th>Company</th>
                    <th>Accountant</th>
                    <th>Date</th>
                    <th>Certificate</th>
                    <th>BOJ5</th>
                    <th>ISIC</th>
                    <th>Loan Agreement</th>
                        <%
                            if (login.isAdmin()){
```

```
                        %>
                      <th>Action</th>
                        <%
                          }
                        %>
                    </tr>
                  </thead>
                  <tbody>
                    <%
                      ComRegDAO comregDAO = new ComRegDAO();
                      LinkedList<CompanyRegistration> comregList =
comregDAO.getAllComReg();
                      for(CompanyRegistration comreg : comregList) {

if(comreg.getCompany().equals(CompanyDAO.getCompanyByID(Integer.parseInt(request.g
etParameter("id"))).getName())) {
                    %>
                    <tr>
                      <td><%= comreg.getCompany() %></td>
                      <td><%= comreg.getAccountant().getUsername()%></td>
                      <td><%= comreg.getDate()%></td>
                      <td><a href="getFiles?fileName=<%=
comreg.getCertificate()%>">Download</a></td>
                      <td><a href="getFiles?fileName=<%=
comreg.getBoj5()%>">Download</a></td>
                      <td><a href="getFiles?fileName=<%=
comreg.getIsic()%>">Download</a></td>
```

```
                      <td><a href="getFiles?fileName=<%=
comreg.getLoanAgreement()%>">Download</a></td>

                      <%
                        if (login.isAdmin()){
                      %>
                      <td class="candidate-list-favourite-time text-right">
                        <form action="DeleteComReg">
                          <button class="btn btn-danger btn-sm" data-toggle="tooltip"
title="Delete"
                              name="id" value="<%= comreg.getID() %>">
                            <i class="far fa-trash-alt"></i>
                          </button>
                        </form>
                      </td>
                      <%
                        }
                      %>
                    </tr>
                    <%
                        }
                      }
                    %>
                  </tbody>
                </table>
              </div>
            </div>
          </div>
```

```
                </main>
              </div>
            </div>
            <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>
            <script src="js/scripts.js"></script>
            <script
src="https://cdn.jsdelivr.net/npm/simple-datatables@7.1.2/dist/umd/simple-datatables.min.js"
crossorigin="anonymous"></script>
            <script src="js/datatables-simple-demo.js"></script>
            <script src="js/datatables-simple-demo_1.js"></script>
        </body>
</html>

<%@page import="OOP.Company"%>
<%@page import="OOP.CompanyDAO"%>
<%@page import="OOP.CompanyRegistration"%>
<%@page import="OOP.ComRegDAO"%>
<%@page import="OOP.User"%>
<%@page import="OOP.UserDAO"%>
<%@page import="java.util.ArrayList" %>
<%@page import="java.util.LinkedList" %>
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
/>
    <meta name="description" content="" />
    <meta name="author" content="" />
    <title>Dashboard</title>
    <link href="https://cdn.jsdelivr.net/npm/simple-datatables@7.1.2/dist/style.min.css"
rel="stylesheet" />
    <link href="css/bootstrap.css" rel="stylesheet" />
    <link href="css/bootstrap-grid.css" rel="stylesheet" />
    <link href="css/bootstrap-reboot.css" rel="stylesheet" />
    <link href="css/Home.css" rel="stylesheet" />
    <script src="https://use.fontawesome.com/releases/v6.3.0/js/all.js"
crossorigin="anonymous"></script>
  </head>
  <body class="sb-nav-fixed">
    <nav class="sb-topnav navbar navbar-expand navbar-dark bg-dark">
      <!-- Navbar Brand-->
      <a class="navbar-brand ps-3" href="index2.html">Pattara</a>

      <!-- Sidebar Toggle-->
      <button class="btn btn-link btn-sm order-1 order-lg-0 me-4 me-lg-0"
id="sidebarToggle"><i class="fas fa-bars"></i></button>
      <!-- Navbar-->
      <ul class="navbar-nav ms-auto me-4">
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" id="navbarDropdown" href="#"
role="button" data-bs-toggle="dropdown" aria-expanded="false"><i class="fas fa-user
fa-fw"></i></a>
```

```
          <ul class="dropdown-menu dropdown-menu-end"
aria-labelledby="navbarDropdown">
            <li><a class="dropdown-item" href="index.html">Logout</a></li>
          </ul>
        </li>
      </ul>
    </nav>
    <div id="layoutSidenav">
      <div id="layoutSidenav_nav">
        <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
          <div class="sb-sidenav-menu">
            <div class="nav">
              <div class="sb-sidenav-menu-heading">Core</div>
              <a class="nav-link" href="index2.jsp">
                <div class="sb-nav-link-icon"><i class="fas fa-tachometer-alt"></i></div>
                Dashboard
              </a>
              <div class="sb-sidenav-menu-heading">Function</div>
              <a class="nav-link collapsed" href="#" data-bs-toggle="collapse"
data-bs-target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">
                <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
                Add
                <div class="sb-sidenav-collapse-arrow"><i class="fas
fa-angle-down"></i></div>
              </a>
              <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne"
data-bs-parent="#sidenavAccordion">
                <nav class="sb-sidenav-menu-nested nav">
```

```
                                    <a class="nav-link" href="addCompany.jsp">Add company</a>
                                    <a class="nav-link" href="addComReg.jsp">Add company
registration</a>
                                    <a class="nav-link" href="addStatement.jsp">Add financial
statement</a>
                                </nav>
                            </div>

                            <div class="sb-sidenav-menu-heading">Manage</div>
                            <a class="nav-link" href="User.jsp">
                                <div class="sb-nav-link-icon"><i class="fas fa-table"></i></div>
                                Users
                            </a>
                        </div>
                    </div>
                </nav>
            </div>
            <!-- table idea from
https://www.tutorialrepublic.com/codelab.php?topic=bootstrap&file=user-management-data-t
able -->
            <div id="layoutSidenav_content">
                <main>
                    <div class="container-fluid px-4">
                        <h1 class="mt-4">Manage User</h1>

                        <div class="card mb-4">
                            <div class="card-header">
                                <i class="fas fa-table me-1"></i>
                                User List
                            </div>
                            <div class="card-body">
                                <table id="datatablesSimple">
                                    <thead>
                                        <tr>
                                            <th>Username</th>
                                            <th>First name</th>
                                            <th>Last name</th>
                                            <th>Role</th>
                                            <th>Action</th>
                                        </tr>
                                    </thead>
                                    <tbody>
<%
    UserDAO userDAO = new UserDAO();
    ArrayList<User> userList = userDAO.getAllUsers();
    for(User user : userList) {
%>
                                        <tr>
                                            <td><%= user.getUsername() %></td>
                                            <td><%= user.getFirstName()%></td>
                                            <td><%= user.getLastName()%></td>
                                            <td><%= user.getAccess()%></td>
                                            <td class="candidate-list-favourite-time text-right">
                                                <div class="btn-group" role="group">
                                                    <form action="UserEdit.jsp">
                            <button class="btn btn-info btn-sm me-1"
data-toggle="tooltip" title="Edit" type="submit"
                              name="username" value="<%= user.getUsername()
%>">
                              <i class="fas fa-cog"></i>
                            </button>
                          </form>

                          <form action="DeleteUser">
                            <button class="btn btn-danger btn-sm"
data-toggle="tooltip" title="Delete"
                              name="username" value="<%= user.getUsername()
%>">
                              <i class="far fa-trash-alt"></i>
                            </button>
                          </form>
                        </div>
                      </td>
                    </tr>
<%
  }
%>
                  </tbody>
                </table>
              </div>
            </div>

          </div>
```

```
                </main>
            </div>
        </div>
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>
        <script src="js/scripts.js"></script>
        <script
src="https://cdn.jsdelivr.net/npm/simple-datatables@7.1.2/dist/umd/simple-datatables.min.js"
crossorigin="anonymous"></script>
        <script src="js/datatables-simple-demo.js"></script>

    </body>
</html>



<%@page import="OOP.User"%>
<%@page import="OOP.UserDAO"%>
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
/>
        <meta name="description" content="" />
        <meta name="author" content="" />
        <title>Dashboard</title>
        <link href="https://cdn.jsdelivr.net/npm/simple-datatables@7.1.2/dist/style.min.css"
rel="stylesheet" />
        <link href="css/bootstrap.css" rel="stylesheet" />
        <link href="css/bootstrap-grid.css" rel="stylesheet" />
        <link href="css/bootstrap-reboot.css" rel="stylesheet" />
        <link href="css/Home.css" rel="stylesheet" />
        <script src="https://use.fontawesome.com/releases/v6.3.0/js/all.js"
crossorigin="anonymous"></script>
    </head>
    <body class="sb-nav-fixed">
        <nav class="sb-topnav navbar navbar-expand navbar-dark bg-dark">
            <!-- Navbar Brand-->
            <a class="navbar-brand ps-3" href="index2.html">Pattara</a>

            <!-- Sidebar Toggle-->
            <button class="btn btn-link btn-sm order-1 order-lg-0 me-4 me-lg-0"
id="sidebarToggle"><i class="fas fa-bars"></i></button>
            <!-- Navbar-->
            <ul class="navbar-nav ms-auto me-4">
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle" id="navbarDropdown" href="#"
role="button" data-bs-toggle="dropdown" aria-expanded="false"><i class="fas fa-user
fa-fw"></i></a>
                    <ul class="dropdown-menu dropdown-menu-end"
aria-labelledby="navbarDropdown">
                        <li><a class="dropdown-item" href="index.html">Logout</a></li>
                    </ul>
                </li>
```

```
      </ul>
    </nav>
    <div id="layoutSidenav">
      <div id="layoutSidenav_nav">
        <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
          <div class="sb-sidenav-menu">
            <div class="nav">
              <div class="sb-sidenav-menu-heading">Core</div>
              <a class="nav-link" href="index2.jsp">
                <div class="sb-nav-link-icon"><i class="fas fa-tachometer-alt"></i></div>
                Dashboard
              </a>
              <div class="sb-sidenav-menu-heading">Function</div>
              <a class="nav-link collapsed" href="#" data-bs-toggle="collapse"
data-bs-target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">
                <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
                Add
                <div class="sb-sidenav-collapse-arrow"><i class="fas
fa-angle-down"></i></div>
              </a>
              <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne"
data-bs-parent="#sidenavAccordion">
                <nav class="sb-sidenav-menu-nested nav">
                  <a class="nav-link" href="addCompany.jsp">Add company</a>
                  <a class="nav-link" href="addComReg.jsp">Add company
registration</a>
                  <a class="nav-link" href="addStatement.jsp">Add financial
statement</a>
```

```
                </nav>
              </div>

              <div class="sb-sidenav-menu-heading">Manage</div>
              <a class="nav-link" href="User.jsp">
                <div class="sb-nav-link-icon"><i class="fas fa-table"></i></div>
                Users
              </a>
            </div>
          </div>
        </nav>
      </div>
      <div id="layoutSidenav_content">
        <main>
          <div class="container-fluid px-4">
            <h1 class="mt-4">Manage User</h1>

            <div class="card mb-4">
              <div class="card-header">
                <i class="fas fa-table me-1"></i>
                Edit User
              </div>
              <div class="card-body">
                  <form accept-charset="UTF-8" action="UserEdit" method="POST"
enctype=application/x-www-form-urlencoded">

                    <div class="form-group">
                      <div class="form-group">
```

```
                        <label for="username"><b>Username</b></label>
                        <input type="text" readonly class="form-control-plaintext"
id="username" name="username" value="<%=request.getParameter("username")%>">
                    </div>

                    <div class="form-group">
                        <label for="firstname"><b>First name</b></label>
                        <input type="text" readonly class="form-control-plaintext"
id="firstname"
value="<%=UserDAO.getFirstNameByUsername(request.getParameter("username"))%>">
                    </div>

                    <div class="form-group">
                        <label for="lastname"><b>Last name</b></label>
                        <input type="text" readonly class="form-control-plaintext"
id="lastname" value="<%=
UserDAO.getLastNameByUsername(request.getParameter("username")) %>">
                    </div>

                    <div class="form-group">
                        <label for="access" class =" col-form-label"><b>Select Access
Level</b></label>
                        <select name="access" class="form-control">
                            <option><%=
UserDAO.getAccessByUsername(request.getParameter("username")) %></option>
                            <option> Admin </option>
                            <option> Manager </option>
                            <option> Viewer </option>
```

```
                                    </select>
                                </div>


                                <button type="submit" class="btn btn-primary">Update</button>
                            </div>
                        </form>


                </div>


            </div>
        </main>
    </div>
</div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>
<script src="js/scripts.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/simple-datatables@7.1.2/dist/umd/simple-datatables.min.js"
crossorigin="anonymous"></script>
<script src="js/datatables-simple-demo.js"></script>

  </body>
</html>


<%--
  Document   : addComReg
  Created on : Jan 4, 2024, 12:07:33?PM
  Author     : IngIng
--%>
<%@page import="OOP.Company"%>
<%@page import="OOP.JOconnection"%>
<%@page import="OOP.CompanyDAO"%>
<%@page import="OOP.UserDAO"%>
<%@page import="OOP.User"%>
<%@page import="java.util.ArrayList" %>
<%@ page language = "java" contentType="text/html; charsetISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<!-- Form template from https://getbootstrap.com/docs/4.0/components/forms/-->
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" href="css/bootstrap.css">
    <link rel="stylesheet" href="css/bootstrap-grid.css">
    <link rel="stylesheet" href="css/bootstrap-reboot.css">
    <style>
      h1 {
        font-size: 20px;
        margin-top: 24px;
        margin-bottom: 24px;
       }

       img {
       height: 60px;
       }
    </style>
  </head>
  <body>
    <div class="col-md-6 offset-md-3 mt-5">
    <h1>Add Company Registration</h1>
    <form action="UploadComReg" method="post" enctype="multipart/form-data">

     <div class="form-group">
      <label for="FormControlSelect1">Select Company</label>
      <select class="form-control" id="FormControlSelect1" name="Company"
required="required">
        <option>Select</option>
<%
  CompanyDAO companyDAO = new CompanyDAO();
  ArrayList<Company> companyList = companyDAO.getAllCompanies();
  for(Company company : companyList) {
%>
        <option><%= company.getName() %></option>
<%
  }
%>
      </select>
     </div>

     <div class="form-group">
      <label for="FormControlSelect2">Select Accountant</label>
      <select class="form-control" id="FormControlSelect2" name="Accountant"
required="required">
```

```
        <option>Select</option>
<%
  UserDAO userDAO = new UserDAO();
  ArrayList<User> userList = userDAO.getAllUsers();
  for(User user : userList) {
%>
        <option><%= user.getUsername() %></option>
<%
  }
%>
      </select>
     </div>

     <div class="form-group">
      <label for="Date" required="required">Date</label>
      <input type="date" name="date" class="form-control" id="Date">
     </div>

     <hr>
     <div class="form-group mt-3">
      <label class="mr-2">Upload Certificate:</label>
      <input type="file" name="Certificate">
     </div>
     <hr>

     <hr>
     <div class="form-group mt-3">
      <label class="mr-2">Upload BOJ5:</label>
      <input type="file" name="BOJ5">
     </div>
     <hr>

     <hr>
     <div class="form-group mt-3">
      <label class="mr-2">Upload ISIC:</label>
      <input type="file" name="ISIC">
     </div>
     <hr>

     <hr>
     <div class="form-group mt-3">
      <label class="mr-2">Upload Loan Agreement:</label>
      <input type="file" name="LoanAgreement">
     </div>
     <hr>

     <button type="submit" class="btn btn-primary">Submit</button>
    </form>
  </div>

  </body>
</html>

<%--
  Document   : addCompany
  Created on : Jan 4, 2024, 12:26:35 PM
```

```
    Author     : IngIng
--%>


<%@page import="OOP.JOconnection"%>
<%@page import="OOP.UserDAO"%>
<%@page import="OOP.User"%>
<%@page import="java.util.ArrayList" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<!-- Form template from https://getbootstrap.com/docs/4.0/components/forms/-->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <link rel="stylesheet" href="css/bootstrap.css">
        <link rel="stylesheet" href="css/bootstrap-grid.css">
        <link rel="stylesheet" href="css/bootstrap-reboot.css">
    </head>
    <body>
        <div class="col-md-6 offset-md-3 mt-5">
            <h1>Add Company</h1>
            <form accept-charset="UTF-8" action="UploadCompany" method="POST"
enctype="multipart/form-data" target="_blank">
                <div class="form-group">
                    <label for="Name" required="required">Company Name</label>
                    <input type="text" name="name" class="form-control" id="Name"
placeholder="Enter company name" >
                </div>
                <div class="form-group">
                    <label for="FormControlSelect1">Select Employee</label>
                    <select class="form-control" id="FormControlSelect1" name="employee" required="required">
                        <option>Select</option>
                        <%
                            UserDAO userDAO = new UserDAO();
                            ArrayList<User> userList = userDAO.getAllUsers();
                            for(User user : userList) {
                        %>
                        <option><%= user.getUsername() %></option>
                        <%
                            }
                        %>
                    </select>
                </div>

                <div class="form-group">
                    <label for="PhoneNum" required="required">Phone Number</label>
                    <input type="text" name="phoneNum" class="form-control" id="PhoneNum" placeholder="Enter phone number">
                </div>

                <button type="submit" class="btn btn-primary">Submit</button>
            </form>
        </div>
    </body>
</html>
```

```
<%--
    Document   : addStatement
    Created on : Jan 4, 2024, 12:07:33 PM
    Author     : IngIng
--%>
<%@page import="OOP.Company"%>
<%@page import="OOP.JOconnection"%>
<%@page import="OOP.CompanyDAO"%>
<%@page import="OOP.User"%>
<%@page import="java.util.ArrayList" %>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<!-- Form template from https://getbootstrap.com/docs/4.0/components/forms/-->
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <link rel="stylesheet" href="css/bootstrap.css">
        <link rel="stylesheet" href="css/bootstrap-grid.css">
        <link rel="stylesheet" href="css/bootstrap-reboot.css">
        <style>
            h1 {
                font-size: 20px;
                margin-top: 24px;
                margin-bottom: 24px;
              }

              img {
```

```
            height: 60px;
            }
        </style>
    </head>
    <body>
        <div class="col-md-6 offset-md-3 mt-5">
        <h1>Add Financial Statement</h1>
        <form accept-charset="UTF-8" action="UploadStatement" method="POST"
enctype="multipart/form-data" target="_blank">

          <div class="form-group">
            <label for="FormControlSelect1">Select Company</label>
            <select class="form-control" id="FormControlSelect1" name="Company"
required="required">
                <option>Select</option>
<%
    CompanyDAO companyDAO = new CompanyDAO();
    ArrayList<Company> companyList = companyDAO.getAllCompanies();
    for(Company company : companyList) {
%>
                <option><%= company.getName() %></option>
<%
    }
%>
            </select>
          </div>

          <div class="form-group">
```

```
      <label for="FormControlSelect2">Select Year</label>
      <select class="form-control" id="FormControlSelect2" name="Year"
required="required">
        <option>Select</option>
<%
  for(int i = 2000; i < 2100; i++) {
%>
        <option><%= i %></option>
<%
  }
%>
      </select>
     </div>

     <hr>
     <div class="form-group mt-3">
      <label class="mr-2">Upload Statement.Excel:</label>
      <input type="file" name="statementExcel">
     </div>
     <hr>

     <hr>
     <div class="form-group mt-3">
      <label class="mr-2">Upload Statement.PDF:</label>
      <input type="file" name="statementPDF">
     </div>
     <hr>
```

```html
          <hr>
          <div class="form-group mt-3">
            <label class="mr-2">Upload VAT Summary:</label>
            <input type="file" name="VAT">
          </div>
          <hr>

          <hr>
          <div class="form-group mt-3">
            <label class="mr-2">Upload PND50:</label>
            <input type="file" name="PND">
          </div>
          <hr>

          <hr>
          <div class="form-group mt-3">
            <label class="mr-2">Upload ProfitAndInterest.Excel:</label>
            <input type="file" name="profit">
          </div>
          <hr>

          <button type="submit" class="btn btn-primary">Submit</button>
        </form>
    </div>
    </body>
</html>
```

```
<!DOCTYPE html>
<!-- Login template form
https://github.com/y2kstack/Archives/blob/b8fe4d0b87efd27199828ffadb8d2b2622100ca5/_
posts/2019-01-27-Project-part2-login-page.md -->
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">

<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;

  width: 100%;

  text-align: center;

  display: inline-block;

}


button:hover {

  opacity: 0.8;

}


/* Center the image and position the close button */

.imgcontainer {

  text-align: center;

  margin: 24px 0 12px 0;

  position: relative;

}


img.avatar {

  width: 40%;

  border-radius: 50%;

}


.container {

  padding: 16px;

}


span.psw {

  float: right;
  padding-top: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
  padding-top: 60px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}

/* The Close Button (x) */
.close {
  position: absolute;
  right: 25px;
  top: 0;
  color: #000;
  font-size: 35px;
  font-weight: bold;
}

.close:hover,
.close:focus {
  color: red;
  cursor: pointer;
}

/* Add Zoom Animation */
.animate {
  -webkit-animation: animatezoom 0.6s;
  animation: animatezoom 0.6s
}

@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)}
```

```
}

</style>
</head>
<body>

<h2>Login Form</h2>



<div>

  <form class="modal-content animate" action="login.jsp">

    <div class="container">
      <label for="username"><b>Username</b></label>
      <input type="text" placeholder="Enter Username" name="username" required>

      <label for="password"><b>Password</b></label>
      <input type="text" placeholder="Enter Password" name="password" required>

      <button type="submit">Login</button>
    </div>

      <div class="container" style="background-color:#f1f1f1; display: flex; justify-content:
space-between;">
      <a href="http://localhost:8089/ComSciIA1/signup.html" class="link"> Doesn't have an
account yet? Signup </a>
```

```
    </div>
  </form>
</div>

<script>
// Get the modal
var modal = document.getElementById('id01');

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
  }
</script>

</body>
</html>

<%@page import="OOP.Company"%>
<%@page import="OOP.CompanyDAO"%>
<%@page import="OOP.CompanyRegistration"%>
<%@page import="OOP.ComRegDAO"%>
<%@page import="OOP.User"%>
<%@page import="java.util.ArrayList" %>
<%@page import="java.util.LinkedList" %>
<!DOCTYPE html>
<!-- Dashboard template from https://startbootstrap.com/template/sb-admin -->
```

```
<html lang="en">
    <head>
        <meta charset="utf-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"
/>
        <meta name="description" content="" />
        <meta name="author" content="" />
        <title>Dashboard</title>
        <link href="https://cdn.jsdelivr.net/npm/simple-datatables@7.1.2/dist/style.min.css"
rel="stylesheet" />
        <link href="css/bootstrap.css" rel="stylesheet" />
        <link href="css/bootstrap-grid.css" rel="stylesheet" />
        <link href="css/bootstrap-reboot.css" rel="stylesheet" />
        <link href="css/Home.css" rel="stylesheet" />
        <script src="https://use.fontawesome.com/releases/v6.3.0/js/all.js"
crossorigin="anonymous"></script>
    </head>
    <body class="sb-nav-fixed">
        <nav class="sb-topnav navbar navbar-expand navbar-dark bg-dark">
            <!-- Navbar Brand-->
            <a class="navbar-brand ps-3" href="index2.html">Pattara</a>

            <!-- Sidebar Toggle-->
            <button class="btn btn-link btn-sm order-1 order-lg-0 me-4 me-lg-0"
id="sidebarToggle"><i class="fas fa-bars"></i></button>
            <!-- Navbar-->
            <ul class="navbar-nav ms-auto me-4">
```

```
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle" id="navbarDropdown" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false"><i class="fas fa-user fa-fw"></i></a>
                    <ul class="dropdown-menu dropdown-menu-end" aria-labelledby="navbarDropdown">
                        <li><a class="dropdown-item" href="index.html">Logout</a></li>
                    </ul>
                </li>
            </ul>
        </nav>
        <div id="layoutSidenav">
            <div id="layoutSidenav_nav">
                <nav class="sb-sidenav accordion sb-sidenav-dark" id="sidenavAccordion">
                    <div class="sb-sidenav-menu">
                        <div class="nav">
                            <div class="sb-sidenav-menu-heading">Core</div>
                            <a class="nav-link" href="index2.jsp">
                                <div class="sb-nav-link-icon"><i class="fas fa-tachometer-alt"></i></div>
                                Dashboard
                            </a>

<%
    User login = (User) session.getAttribute("login");
    if (login.isManager() || login.isAdmin()){
%>
                            <div class="sb-sidenav-menu-heading">Function</div>
```

```
              <a class="nav-link collapsed" href="#" data-bs-toggle="collapse"
data-bs-target="#collapseLayouts" aria-expanded="false" aria-controls="collapseLayouts">
                <div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
                Add
                <div class="sb-sidenav-collapse-arrow"><i class="fas
fa-angle-down"></i></div>
              </a>
              <div class="collapse" id="collapseLayouts" aria-labelledby="headingOne"
data-bs-parent="#sidenavAccordion">
                <nav class="sb-sidenav-menu-nested nav">
                  <a class="nav-link" href="addCompany.jsp">Add company</a>
                  <a class="nav-link" href="addComReg.jsp">Add company
registration</a>
                  <a class="nav-link" href="addStatement.jsp">Add financial
statement</a>
                </nav>
              </div>
<%
  }
  if (login.isAdmin()){
%>
              <div class="sb-sidenav-menu-heading">Manage</div>
              <a class="nav-link" href="User.jsp">
                <div class="sb-nav-link-icon"><i class="fas fa-table"></i></div>
                Users
              </a>
<%
  }
```

```
%>
            </div>
          </div>
        </nav>
      </div>
      <div id="layoutSidenav_content">
        <main>
          <div class="container-fluid px-4">
            <h1 class="mt-4">Dashboard</h1>

<%
  if (login.isManager() || login.isAdmin()){
%>
            <div class="row">
              <div class="col">
                <div class="card bg-primary text-white mb-4">
                  <div class="card-body">Add company</div>
                  <div class="card-footer d-flex align-items-center
justify-content-between">
                    <a class="small text-white stretched-link"
href="addCompany.jsp">Go</a>
                    <div class="small text-white"><i class="fas
fa-angle-right"></i></div>
                  </div>
                </div>
              </div>
              <div class="col">
                <div class="card bg-warning text-white mb-4">
```

```
                                    <div class="card-body">Add company registration</div>
                                    <div class="card-footer d-flex align-items-center
justify-content-between">
                                        <a class="small text-white stretched-link"
href="addComReg.jsp">Go</a>
                                        <div class="small text-white"><i class="fas
fa-angle-right"></i></div>
                                    </div>
                                </div>
                            </div>
                            <div class="col">
                                <div class="card bg-success text-white mb-4">
                                    <div class="card-body">Add financial statement</div>
                                    <div class="card-footer d-flex align-items-center
justify-content-between">
                                        <a class="small text-white stretched-link"
href="addStatement.jsp">Go</a>
                                        <div class="small text-white"><i class="fas
fa-angle-right"></i></div>
                                    </div>
                                </div>
                            </div>
                        </div>
<%
    }
%>

                        <div class="card mb-4">
```

```
              <div class="card-header">
                <i class="fas fa-table me-1"></i>
                Company List
              </div>
              <div class="card-body">
                <table id="datatablesSimple">
                  <thead>
                    <tr>
                      <th>Name</th>
                      <th>Employee In Charge</th>
                      <th>Phone Number</th>
                      <th>Action</th>
                    </tr>
                  </thead>
                  <tbody>
<%
  CompanyDAO companyDAO = new CompanyDAO();
  ArrayList<Company> companyList = companyDAO.getAllCompanies();
  for(Company company : companyList) {
%>
                    <tr>
                      <td><%= company.getName() %></td>
                      <td><%=
company.getEmployeeInCharge().getUsername()%></td>
                      <td><%= company.getPhoneNumber()%></td>
                      <td class="candidate-list-favourite-time text-right">
                        <div class="btn-group" role="group">
                          <form action="CompanyView.jsp">
```

```
                            <button class="btn btn-info btn-sm me-1"
data-toggle="tooltip" title="View" type="submit"
                              name="id" value="<%= company.getID() %>">
                              <i class="far fa-eye"></i>
                            </button>
                          </form>
<%
  if (login.isAdmin()){
%>
                          <form action="DeleteCompany">
                            <button class="btn btn-danger btn-sm"
data-toggle="tooltip" title="Delete"
                              name="id" value="<%= company.getID() %>">
                              <i class="far fa-trash-alt"></i>
                            </button>
                          </form>
<%
  }
%>
                        </div>
                      </td>
                    </tr>
<%
  }
%>
                  </tbody>
                </table>
              </div>
```

```
                        </div>

                        <div class="card mb-4">
                            <div class="card-header">
                                <i class="fas fa-table me-1"></i>
                                Company Registration List
                            </div>
                            <div class="card-body">
                                <table id="datatablesSimple1">
                                    <thead>
                                        <tr>
                                            <th>Company</th>
                                            <th>Accountant</th>
                                            <th>Date</th>
                                            <th>Certificate</th>
                                            <th>BOJ5</th>
                                            <th>ISIC</th>
                                            <th>Loan Agreement</th>
<%
    if (login.isAdmin()){
%>
                                            <th>Action</th>
<%
    }
%>
                                        </tr>
                                    </thead>
                                    <tbody>
```

```
<%
    ComRegDAO comregDAO = new ComRegDAO();
    LinkedList<CompanyRegistration> comregList = comregDAO.getAllComReg();
    for(CompanyRegistration comreg : comregList) {
%>
                                        <tr>
                                            <td><%= comreg.getCompany() %></td>
                                            <td><%= comreg.getAccountant().getUsername()%></td>
                                            <td><%= comreg.getDate()%></td>
                                            <td><a href="getFiles?fileName=<%=
comreg.getCertificate()%>">Download</a></td>
                                            <td><a href="getFiles?fileName=<%=
comreg.getBoj5()%>">Download</a></td>
                                            <td><a href="getFiles?fileName=<%=
comreg.getIsic()%>">Download</a></td>
                                            <td><a href="getFiles?fileName=<%=
comreg.getLoanAgreement()%>">Download</a></td>
<%
    if (login.isAdmin()){
%>
                                            <td class="candidate-list-favourite-time text-right">
                                                    <form action="DeleteComReg">
                                                        <button class="btn btn-danger btn-sm"
data-toggle="tooltip" title="Delete"
                                                            name="id" value="<%= comreg.getID() %>">
                                                            <i class="far fa-trash-alt"></i>
                                                        </button>
                                                    </form>
                                        </td>
<%
    }
%>
                                    </tr>
<%
    }
%>
                                </tbody>
                            </table>
                        </div>
                    </div>
                </div>
            </main>
        </div>
    </div>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
crossorigin="anonymous"></script>
    <script src="js/scripts.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/simple-datatables@7.1.2/dist/umd/simple-datatables.min.js"
crossorigin="anonymous"></script>
    <script src="js/datatables-simple-demo.js"></script>
    <script src="js/datatables-simple-demo_1.js"></script>

  </body>
</html>
```

```
<%@page import="OOP.JOconnection"%>
<%@page import="OOP.UserDAO"%>
<%@page import="OOP.User"%>
<%@page language="java" import="java.sql.*"%>
<!-- Login template form
https://github.com/y2kstack/Archives/blob/b8fe4d0b87efd27199828ffadb8d2b2622100ca5/_
posts/2019-01-27-Project-part2-login-page.md -->
<%
try {
   String Username = request.getParameter("username");
   String Password = request.getParameter("password");

   JOconnection.JConnect();

   // Use a prepared statement to avoid SQL injection
   String selectQuery = "SELECT * FROM register WHERE username=? AND password=?";

   try (PreparedStatement pstmt = JOconnection.conn.prepareStatement(selectQuery)) {
      pstmt.setString(1, Username);
      pstmt.setString(2, Password);

      JOconnection.rslt = pstmt.executeQuery();

      if (JOconnection.rslt.next()) {
      User Login = UserDAO.getUserByName(Username);
      session.setAttribute("login", Login);
%>
<script>
    window.location.href="http://localhost:8089/ComSciIA1/index2.jsp";
</script>
<%
        } else {
%>
<script>
    alert("Your username or password is wrong. Please try again.");
    window.location.href="http://localhost:8089/ComSciIA1/index.html";
</script>
<%
        }
    }
} catch (ClassNotFoundException | SQLException e) {
    e.printStackTrace();
}
%>
```

```
<!DOCTYPE html>
<!-- Signup template form
https://github.com/y2kstack/Archives/blob/b8fe4d0b87efd27199828ffadb8d2b2622100ca5/_
posts/2019-01-26-Project-part1-signuppage.md -->
<html>
<style>
body {font-family: Arial, Helvetica, sans-serif;}
* {box-sizing: border-box;}
```

```
/* Full-width input fields */

input[type=text], input[type=password] {

  width: 100%;

  padding: 15px;

  margin: 5px 0 22px 0;

  display: inline-block;

  border: none;

  background: #f1f1f1;

}


/* Add a background color when the inputs get focus */

input[type=text]:focus, input[type=password]:focus {

  background-color: #ddd;

  outline: none;

}


/* Set a style for all buttons */

button {

  background-color: #4CAF50;

  color: white;

  padding: 14px 20px;

  margin: 8px 0;

  border: none;

  cursor: pointer;

  width: 100%;

  opacity: 0.9;

}
```

```
.h2{
   text-align: center;
}

button:hover {
  opacity:1;
}

/* Extra styles for the cancel button */
.cancelbtn {
  padding: 14px 20px;
  background-color: #f44336;
}

/* Float cancel and signup buttons and add an equal width */
.cancelbtn, .signupbtn {
  float: left;
  width: 50%;
}

/* Add padding to container elements */
.container {
  padding: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: #474e5d;
  padding-top: 50px;
}

/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}

/* Style the horizontal ruler */
hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}

/* The Close Button (x) */
.close {
  position: absolute;
  right: 35px;
  top: 15px;
  font-size: 40px;
  font-weight: bold;
  color: #f1f1f1;
}

.close:hover,
.close:focus {
  color: #f44336;
  cursor: pointer;
}

/* Clear floats */
.clearfix::after {
  content: "";
  clear: both;
  display: table;
}

/* Change styles for cancel button and signup button on extra small screens */
@media screen and (max-width: 300px) {
  .cancelbtn, .signupbtn {
     width: 100%;
  }
}
</style>
```

```
<body>


<h2>Signup Form</h2>


<div>


  <form class="modal-content" action="signup.jsp">
    <div class="container">
      <h1>Sign Up</h1>
      <p>Please fill in this form to create an account.</p>
      <hr>
      <label for="firstname"><b>First Name</b></label>
      <input type="text" placeholder="First Name" name="firstname" required>

      <label for="lastname"><b>Last Name</b></label>
      <input type="text" placeholder="Last Name" name="lastname" required>

      <label for="username"><b>Enter Username</b></label>
      <input type="text" placeholder="Enter Username" name="username" required>

      <label for="password"><b>Enter password</b></label>
      <input type="Password" placeholder="Enter password" name="password" required>

      <label for="email"><b>Enter Email</b></label>
      <input type="text" placeholder="Enter Email" name="email" required>

      <label for="mobileNum"><b>mobile no</b></label>
      <input type="text" placeholder="mobile no" name="mobileNum" required>
```

```
   <label for="access"></label>
   <input type="hidden" value="null" name="access">

   <div class="clearfix">
    <button type="button"
onclick="window.location.href='http://localhost:8089/ComSciIA1/index.html'"
class="cancelbtn">Cancel</button>
    <button type="submit" class="signupbtn">Sign Up</button>
   </div>

   <p>By creating an account you agree to our <a href="#" style="color:dodgerblue">Terms
& Privacy</a>.</p>

  </div>
 </form>
</div>




</body>
</html>
<%@page import="OOP.JOconnection"%>
<%@page language="java" import="java.sql.*"%>
<%@page import="java.io.*"%>
<!--Sign up template from
https://github.com/y2kstack/Archives/blob/b8fe4d0b87efd27199828ffadb8d2b2622100ca5/_
posts/2019-01-26-Project-part1-signuppage.md-->
```

```
<%
    try {
        ResultSet rs = null;
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb",
"root", "li18012006");
        Statement stmt = con.createStatement();
        String firstname = request.getParameter("firstname");
        String lastname = request.getParameter("lastname");
        String username = request.getParameter("username");
        String password = request.getParameter("password");
        String email = request.getParameter("email");
        String mobileNum = request.getParameter("mobileNum");
        String access = request.getParameter("access");

        JOconnection.JConnect();
        String checkUsernameQuery = "SELECT * FROM register WHERE username = ?";
        try (PreparedStatement checkUsernameStmt =
JOconnection.conn.prepareStatement(checkUsernameQuery)) {
            checkUsernameStmt.setString(1, username);
            ResultSet resultSet = checkUsernameStmt.executeQuery();

            if (resultSet.next()) {
                // Username already exists, show an error message using JavaScript
%>
<script>
                alert("Username already exists. Please choose another username.");
                window.location.href = "signup.html";
</script>
<%
            } else {
                stmt.executeUpdate("insert into register values('" + firstname + "','" + lastname +
"','" + username + "','" + password + "','" + email + "','" + mobileNum + "','" + access + "')");
                response.sendRedirect("index.html");
            }
        }

        con.close();
        stmt.close();
    } catch (ClassNotFoundException | SQLException e) {
        out.println(e.getMessage());
    }
%>

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package OOP;

import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.LinkedList;

/**
```

```
 *
 * @author IngIng
 */
public class ComRegDAO {
    public LinkedList<CompanyRegistration> getAllComReg() {
        LinkedList<CompanyRegistration> ComRegList = new LinkedList<>();
        try {
            JOconnection.JConnect(); // Establish connection using JOconnection class

            String selectquery = ("SELECT * FROM companyregistration");
            JOconnection.rslt = JOconnection.stmt.executeQuery(selectquery);

            while (JOconnection.rslt.next()) {
                CompanyRegistration ComReg = new CompanyRegistration(
                        JOconnection.rslt.getString("company"),
                        UserDAO.getUserByName(JOconnection.rslt.getString("accountant")),
                        JOconnection.rslt.getString("date"),
                        JOconnection.rslt.getString("certificate"),
                        JOconnection.rslt.getString("boj5"),
                        JOconnection.rslt.getString("isic"),
                        JOconnection.rslt.getString("loanAgreement"),
                        JOconnection.rslt.getInt("id")
                );

                ComRegList.add(ComReg);
            }
        } catch (SQLException | ClassNotFoundException e) {
            e.printStackTrace();
        } finally {
            try {
                JOconnection.conn.close(); // Close the connection in the finally block
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return ComRegList;
    }
}

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package OOP;

/**
 *
 * @author IngIng
 */
public class Company {

    private String name;
    private User employeeInCharge;
    private String phoneNumber;
    private int ID;
```

```
public Company(){
}

public Company(String name, User employeeInCharge, String phoneNumber, int id) {
    this.name = name;
    this.employeeInCharge = employeeInCharge;
    this.phoneNumber = phoneNumber;
    this.ID = id;
}

public String getName() {
    return name;
}

public User getEmployeeInCharge() {
    return employeeInCharge;
}

public String getPhoneNumber() {
    return phoneNumber;
}

public int getID() {
    return ID;
}

public void setName(String name) {
        this.name = name;
    }

    public void setEmployeeInCharge(User employeeInCharge) {
        this.employeeInCharge = employeeInCharge;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public void setID(int id) {
        this.ID = id;
    }

}

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package OOP;

/**
 *
 * @author IngIng
 */
```

```
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

public class CompanyDAO {

    // Method to retrieve all companies from the database and map to Company objects
    public ArrayList<Company> getAllCompanies() {
        ArrayList<Company> companyList = new ArrayList<>();
        try {
            JOconnection.JConnect(); // Establish connection using JOconnection class

            String selectquery = ("SELECT * FROM company");
            JOconnection.rslt = JOconnection.stmt.executeQuery(selectquery);

            while (JOconnection.rslt.next()) {
                Company companies = new Company(
                        JOconnection.rslt.getString("name"),

UserDAO.getUserByName(JOconnection.rslt.getString("employeeInCharge")),
                        JOconnection.rslt.getString("phoneNumber"),
                        JOconnection.rslt.getInt("id")
                );

                companyList.add(companies);
            }
        } catch (SQLException | ClassNotFoundException e) {
            e.printStackTrace();
        } finally {
            try {
                JOconnection.conn.close(); // Close the connection in the finally block
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return companyList;
    }

    public static String getEmployeeByID(int id) {
        try {
            JOconnection.JConnect();

            String selectQuery = "SELECT EmployeeInCharge FROM company WHERE id = ?";
            try (PreparedStatement statement =
JOconnection.conn.prepareStatement(selectQuery)) {
                statement.setInt(1, id);

                try (ResultSet resultSet = statement.executeQuery()) {
                    if (resultSet.next()) {
                        return resultSet.getString("employeeInCharge");
                    } else {
                        return null;
                    }
                }
            }
        } catch (SQLException | ClassNotFoundException e) {
            e.printStackTrace();
            return null;
        } finally {
            try {
                JOconnection.conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }

    public static String getPhoneByID(int id) {
        try {
            JOconnection.JConnect();

            String selectQuery = "SELECT phoneNumber FROM company WHERE id = ?";
            try (PreparedStatement statement =
JOconnection.conn.prepareStatement(selectQuery)) {
                statement.setInt(1, id);

                try (ResultSet resultSet = statement.executeQuery()) {
                    if (resultSet.next()) {
                        return resultSet.getString("phoneNumber");
                    } else {
                        return null;
                    }
                }
            }
        } catch (SQLException | ClassNotFoundException e) {
            e.printStackTrace();
            return null;
        } finally {
            try {
                JOconnection.conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }

    public static String getNameByID(int id) {
        try {
            JOconnection.JConnect();

            String selectQuery = "SELECT name FROM company WHERE id = ?";
            try (PreparedStatement statement =
JOconnection.conn.prepareStatement(selectQuery)) {
                statement.setInt(1, id);

                try (ResultSet resultSet = statement.executeQuery()) {
                    if (resultSet.next()) {
                        return resultSet.getString("name");
                    } else {
                        return null;
                    }
                }
            }
        } catch (SQLException | ClassNotFoundException e) {
            e.printStackTrace();
            return null;
        } finally {
            try {
                JOconnection.conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }

    public static Company getCompanyByID(int id) {
        Company company = null;
        try {
            JOconnection.JConnect();

            String selectQuery = "SELECT * FROM company WHERE id = ?";
            try (PreparedStatement statement =
JOconnection.conn.prepareStatement(selectQuery)) {
                statement.setInt(1, id);

                try (ResultSet resultSet = statement.executeQuery()) {
                    if (resultSet.next()) {
                        company = new Company();
                        company.setName(resultSet.getString("name"));
company.setEmployeeInCharge(UserDAO.getUserByName(resultSet.getString("employeeIn
Charge")));
                        company.setPhoneNumber(resultSet.getString("phoneNumber"));
                        company.setID(resultSet.getInt("id"));
                    }
                }
            }
        } catch (SQLException | ClassNotFoundException e) {
            e.printStackTrace();
        } finally {
            try {
                JOconnection.conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

        return company;
    }

}

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
```

```
package OOP;

/**
 *
 * @author IngIng
 */
public class CompanyRegistration {
    private String company;
    private User accountant;
    private String date;
    private String certificate;
    private String boj5;
    private String isic;
    private String loanAgreement;
    private int ID;

    // Constructor
    public CompanyRegistration(String company, User accountant, String Date,
                                String certificate, String boj5, String isic, String loanAgreement, int ID) {
        this.company = company;
        this.accountant = accountant;
        this.date = Date;
        this.certificate = certificate;
        this.boj5 = boj5;
        this.isic = isic;
        this.loanAgreement = loanAgreement;
        this.ID = ID;
    }
```

```
// Getter methods

public User getAccountant() {
    return accountant;
}

public String getCompany() {
    return company;
}

public String getDate() {
    return date;
}

public String getCertificate() {
    return certificate;
}

public String getBoj5() {
    return boj5;
}

public String getIsic() {
    return isic;
}

public String getLoanAgreement() {
    return loanAgreement;
}

public int getID() {
    return ID;
}

// Setter methods

public void setAccountant(User accountant) {
    this.accountant = accountant;
}

public void setCompany(String comp) {
    this.company = comp;
}

public void setDate(String Date) {
    this.date = Date;
}

public void setCertificate(String certificate) {
    this.certificate = certificate;
}

public void setBoj5(String boj5) {
    this.boj5 = boj5;
}
```

```
    public void setIsic(String isic) {
        this.isic = isic;
    }

    public void setLoanAgreement(String loanAgreement) {
        this.loanAgreement = loanAgreement;
    }
}

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package OOP;

import jakarta.servlet.RequestDispatcher;
import java.io.IOException;
import java.sql.*;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
/**
 *
 * @author IngIng
 */
```

```
public class DisplayFile extends HttpServlet {
      private static final long serialVersionUID = 1L;


  public DisplayFile() {
    super();
  }
    @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
            response.getWriter().append("Served at:
").append(request.getContextPath());
      }

    @Override
      protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {

            System.out.println("In do post method of Display Image servlet.");
            String imageId=request.getParameter("imageId");
            int id=Integer.parseInt(imageId);

            //getting database connection (jdbc code)
                        Connection connection=null;
                        int imgId=0;
                        String imgFileName=null;
                        try
                        {
                              JOconnection.JConnect();
```

```
                                        String query="select * from image";
                                        JOconnection.stmt = connection.createStatement();
                                        ResultSet rs =
JOconnection.stmt.executeQuery(query);

                                        while(rs.next())
                                        {
                                              if(rs.getInt("imageId")==id)
                                              {
                                                    imgId=rs.getInt("imageId");

imgFileName=rs.getString("imageFileName");
                                              }
                                        }

                                  }
                                  catch (Exception e)
                                  {
                                        System.out.println(e);
                                  }

                                  RequestDispatcher rd;
                                  request.setAttribute("id",String.valueOf(imgId));  //valueOf
                                  request.setAttribute("img",imgFileName);
                                  rd=request.getRequestDispatcher("displayImage.jsp");
                                  rd.forward(request, response);

      }
```

```
}

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package OOP;

/**
 *
 * @author IngIng
 */
public class FinancialStatement {
    private String company;
    private int year;
    private String excelStatement;
    private String vatSummary;
    private String pdfStatement;
    private String pnd50;
    private String excelProfitAndInterest;
    private int ID;

    public FinancialStatement(String company, int year, String excelStatement, String
vatSummary, String pdfStatement, String pnd50, String excelProfitAndInterest, int ID) {
        this.company = company;
        this.year = year;
    this.excelStatement = excelStatement;
    this.vatSummary = vatSummary;
    this.pdfStatement = pdfStatement;
    this.pnd50 = pnd50;
    this.excelProfitAndInterest = excelProfitAndInterest;
    this.ID = ID;
}

// Getter methods
public String getCompany() {
    return company;
}

public int getYear() {
    return year;
}

public String getExcelStatement() {
    return excelStatement;
}

public String getVatSummary() {
    return vatSummary;
}

public String getPdfStatement() {
    return pdfStatement;
}
```

```
public String getPnd50() {

    return pnd50;

}


public String getExcelProfitAndInterest() {

    return excelProfitAndInterest;

}


public int getID(){

    return ID;

}


// Setter methods

public void setYear(int year) {

    this.year = year;

}


public void setExcelStatement(String excelStatement) {

    this.excelStatement = excelStatement;

}


public void setVatSummary(String vatSummary) {

    this.vatSummary = vatSummary;

}


public void setPdfStatement(String pdfStatement) {

    this.pdfStatement = pdfStatement;
    }

    public void setPnd50(String pnd50) {
        this.pnd50 = pnd50;
    }

    public void setExcelProfitAndInterest(String excelProfitAndInterest) {
        this.excelProfitAndInterest = excelProfitAndInterest;
    }
}

package OOP;

import java.sql.*;
import java.util.*;
import java.io.*;
//template from
https://github.com/y2kstack/Archives/blob/b8fe4d0b87efd27199828ffadb8d2b2622100ca5/_
posts/2019-01-27-Project-part2-login-page.md

public class JOconnection {
    public static Connection conn;
    public static Statement stmt;
    public static ResultSet rslt;

    public static void JConnect() throws  ClassNotFoundException, SQLException {
        try{
            String username ="root";
            String password ="li18012006";

            Class.forName("com.mysql.cj.jdbc.Driver");
            conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb",username,password);
            stmt = conn.createStatement();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
        }
}
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package OOP;

import java.sql.SQLException;
import java.util.LinkedList;

/**
 *
 * @author IngIng
 */
public class StatementDAO {
    public LinkedList<FinancialStatement> getAllStatement() {
        LinkedList<FinancialStatement> StatementList = new LinkedList<>();
```

```
try {
    JOconnection.JConnect(); // Establish connection using JOconnection class

    String selectquery = ("SELECT * FROM financialstatement");
    JOconnection.rslt = JOconnection.stmt.executeQuery(selectquery);

    while (JOconnection.rslt.next()) {
        FinancialStatement Statement = new FinancialStatement(
                JOconnection.rslt.getString("company"),
                Integer.parseInt(JOconnection.rslt.getString("year")),
                JOconnection.rslt.getString("excelStatement"),
                JOconnection.rslt.getString("pdfStatement"),
                JOconnection.rslt.getString("vatSummary"),
                JOconnection.rslt.getString("pnd50"),
                JOconnection.rslt.getString("excelProfitAndInterest"),
                JOconnection.rslt.getInt("id")
        );

        StatementList.add(Statement);
    }
} catch (SQLException | ClassNotFoundException e) {
    e.printStackTrace();
} finally {
    try {
        JOconnection.conn.close(); // Close the connection in the finally block
    } catch (SQLException e) {
        e.printStackTrace();
    }
```

```
        }
        return StatementList;
    }
}

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package OOP;

public class User {

    private String firstName;
    private String lastName;
    private String username;
    private String email;
    private String password;
    private String mobileNum;
    private String accessLevel;

    public User() {
    }

    // Constructor with all fields
    public User(String firstName, String lastName, String username, String email, String
password, String mobileNum, String access) {
    this.firstName = firstName;
    this.lastName = lastName;
    this.username = username;
    this.email = email;
    this.password = password;
    this.mobileNum = mobileNum;
    this.accessLevel = access;
}

// Setters
public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public void setLastName(String lastName) {
    this.lastName = lastName;
}

public void setUsername(String username) {
    this.username = username;
}

public void setEmail(String email) {
    this.email = email;
}

public void setPassword(String password) {
    this.password = password;
```

```
}

public void setMobileNum(String mobileNum) {
    this.mobileNum = mobileNum;
}

public void setAccess(String Access) {
    this.accessLevel = Access;
}

// Getters
public String getFirstName() {
    return firstName;
}

public String getLastName() {
    return lastName;
}

public String getUsername() {
    return username;
}

public String getEmail() {
    return email;
}

public String getPassword() {
        return password;
    }

    public String getMobileNum() {
        return mobileNum;
    }

    public String getAccess() {
        return accessLevel;
    }

    public boolean isAdmin() {
        return "Admin".equals(accessLevel);
    }

    public boolean isManager() {
        return "Manager".equals(accessLevel);
    }
}

package OOP; // Replace with your actual package name

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

public class UserDAO {
    public ArrayList<User> getAllUsers() {
        ArrayList<User> userList = new ArrayList<>();

        try {
            JOconnection.JConnect();

            String selectquery = "SELECT * FROM register";
            try (PreparedStatement statement =
JOconnection.conn.prepareStatement(selectquery);
                 ResultSet resultSet = statement.executeQuery()) {

                while (resultSet.next()) {
                    User user = new User();
                    user.setFirstName(resultSet.getString("firstname"));
                    user.setLastName(resultSet.getString("lastname"));
                    user.setUsername(resultSet.getString("username"));
                    user.setEmail(resultSet.getString("email"));
                    user.setPassword(resultSet.getString("password"));
                    user.setMobileNum(resultSet.getString("mobileNum"));
                    user.setAccess(resultSet.getString("access"));

                    userList.add(user);
                }
            }
        } catch (SQLException | ClassNotFoundException e) {
            e.printStackTrace();
        }
        return userList;
    }

    public static User getUserByName(String username) {
        User user = null;

        try {
            JOconnection.JConnect();

            String selectQuery = "SELECT * FROM register WHERE username = ?";
            try (PreparedStatement statement =
JOconnection.conn.prepareStatement(selectQuery)) {
                statement.setString(1, username);

                try (ResultSet resultSet = statement.executeQuery()) {
                    if (resultSet.next()) {
                        user = new User();
                        user.setFirstName(resultSet.getString("firstName"));
                        user.setLastName(resultSet.getString("lastName"));
                        user.setUsername(resultSet.getString("username"));
                        user.setEmail(resultSet.getString("email"));
                        user.setPassword(resultSet.getString("password"));
                        user.setMobileNum(resultSet.getString("mobileNum"));
                        user.setAccess(resultSet.getString("access"));
                    }
                }
            }
        } catch (SQLException | ClassNotFoundException e) {
            e.printStackTrace();
        } finally {
            try {
                JOconnection.conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

        return user;
    }

    public static String getFirstNameByUsername(String user) {
        try {
            JOconnection.JConnect();

            String selectQuery = "SELECT firstname FROM register WHERE username = ?";
            try (PreparedStatement statement =
JOconnection.conn.prepareStatement(selectQuery)) {
                statement.setString(1, user);

                try (ResultSet resultSet = statement.executeQuery()) {
                    if (resultSet.next()) {
                        return resultSet.getString("firstname");
                    } else {
                        return null;
                    }
                }
            }
        } catch (SQLException | ClassNotFoundException e) {
            e.printStackTrace();
            return null;
        } finally {
            try {
                JOconnection.conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }

    public static String getLastNameByUsername(String user) {
        try {
            JOconnection.JConnect();

            String selectQuery = "SELECT lastname FROM register WHERE username = ?";
            try (PreparedStatement statement =
JOconnection.conn.prepareStatement(selectQuery)) {
                statement.setString(1, user);

                try (ResultSet resultSet = statement.executeQuery()) {
                    if (resultSet.next()) {
                        return resultSet.getString("lastname");
                    } else {
                        return null;
                    }
                }
            }
        } catch (SQLException | ClassNotFoundException e) {
            e.printStackTrace();
            return null;
        } finally {
            try {
                JOconnection.conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }

    public static String getAccessByUsername(String user) {
        try {
            JOconnection.JConnect();

            String selectQuery = "SELECT access FROM register WHERE username = ?";
            try (PreparedStatement statement =
JOconnection.conn.prepareStatement(selectQuery)) {
                statement.setString(1, user);

                try (ResultSet resultSet = statement.executeQuery()) {
                    if (resultSet.next()) {
                        return resultSet.getString("access");
                    } else {
                        return null;
                    }
                }
            }
        } catch (SQLException | ClassNotFoundException e) {
            e.printStackTrace();
            return null;
        } finally {
            try {
                JOconnection.conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this template
 * Delete data from mySQL template from https://www.youtube.com/watch?v=dIFIKjozsSc&ab_channel=CSITBAMU
 */
package Servlet;

import OOP.JOconnection;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import static java.lang.System.out;

/**
 *
 * @author IngIng
 */
@WebServlet(name = "DeleteComReg", urlPatterns = {"/DeleteComReg"})
public class DeleteComReg extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
```

```
        response.setContentType("text/html;charset=UTF-8");
        String ID = request.getParameter("id");
        try{
            JOconnection.JConnect();

            String sql = "DELETE from companyregistration WHERE id ="+ID;

            response.sendRedirect("index2.jsp");
            JOconnection.stmt.executeUpdate(sql);
        }
        catch(Exception e){
            out.println(e.toString());
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
```

```
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this
template
 * Delete data from mySQL template from
https://www.youtube.com/watch?v=dIFIKjozsSc&ab_channel=CSITBAMU
 */
package Servlet;

import OOP.JOconnection;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import static java.lang.System.out;

/**
 *
 * @author IngIng
 */
@WebServlet(name = "DeleteCompany", urlPatterns = {"/DeleteCompany"})
public class DeleteCompany extends HttpServlet {
    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String ID = request.getParameter("id");
        try{
            JOconnection.JConnect();

            String sql = "DELETE from company WHERE id ="+ID;

            response.sendRedirect("index2.jsp");
            JOconnection.stmt.executeUpdate(sql);
        }
        catch(Exception e){
            out.println(e.toString());
        }
    }
    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this
template
 * Delete data from mySQL template from
https://www.youtube.com/watch?v=dIFIKjozsSc&ab_channel=CSITBAMU
 */
package Servlet;

import OOP.JOconnection;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import static java.lang.System.out;

/**
 *
 * @author IngIng
 */
@WebServlet(name = "DeleteStatement", urlPatterns = {"/DeleteStatement"})
public class DeleteStatement extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        String ID = request.getParameter("id");
        try{
            JOconnection.JConnect();

            String sql = "DELETE from financialstatement WHERE id ="+ID;

            response.sendRedirect("index2.jsp");
            JOconnection.stmt.executeUpdate(sql);
        }
        catch(Exception e){
            out.println(e.toString());
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on
the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}

package Servlet;
```

```
import OOP.JOconnection;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.nio.file.StandardCopyOption;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.MultipartConfig;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import jakarta.servlet.http.Part;
//Upload file to mySQL code from
https://github.com/Java-Hub18/Servlet-JSP/tree/11ba5e0f8d56be4e8a847ec3d4df1f99623ae
2c0/ServletUploadDownloadFile
@WebServlet(name = "UploadComReg", urlPatterns = {"/UploadComReg"})
@MultipartConfig(fileSizeThreshold = 1024 * 1024 * 10, // 10 MB
        maxFileSize = 1024 * 1024 * 1000, // 1 GB
        maxRequestSize = 1024 * 1024 * 1000)      // 1 GB
public class UploadComReg extends HttpServlet {

    PrintWriter out = null;
    PreparedStatement ps = null;

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/plain;charset=UTF-8");
        try {
            out = response.getWriter();
            request.getSession(false);
            String folderName = "Files";
            String uploadPath = request.getServletContext().getRealPath("") + File.separator +
folderName;
            File dir = new File(uploadPath);
            if (!dir.exists()) {
                dir.mkdirs();
            }
            Part cer = request.getPart("Certificate");
            Part boj = request.getPart("BOJ5");
            Part isic = request.getPart("ISIC");
            Part loan = request.getPart("LoanAgreement");
            String company = request.getParameter("Company");
            String user = request.getParameter("Accountant");
            String date = request.getParameter("date");

            String cerFileName = cer.getSubmittedFileName();
            String bojFileName = boj.getSubmittedFileName();
            String isicFileName = isic.getSubmittedFileName();
            String loanFileName = loan.getSubmittedFileName();
            InputStream isCer = cer.getInputStream();
            InputStream isBOJ = boj.getInputStream();
            InputStream isISIC = isic.getInputStream();
            InputStream isLoan = loan.getInputStream();

            Files.copy(isCer, Paths.get(uploadPath + File.separator + cerFileName),
StandardCopyOption.REPLACE_EXISTING);
            Files.copy(isBOJ, Paths.get(uploadPath + File.separator + bojFileName),
StandardCopyOption.REPLACE_EXISTING);
            Files.copy(isISIC, Paths.get(uploadPath + File.separator + isicFileName),
StandardCopyOption.REPLACE_EXISTING);
            Files.copy(isLoan, Paths.get(uploadPath + File.separator + loanFileName),
StandardCopyOption.REPLACE_EXISTING);

            try {
                JOconnection.JConnect();
                String sql = "insert into companyregistration(company, accountant, date,
certificate, boj5, isic, loanAgreement) values(?, ?, ?, ?, ?, ?, ?)";
                ps=JOconnection.conn.prepareStatement(sql);
                     ps.setString(1,company);
                ps.setString(2,user);
                ps.setString(3,date);
                ps.setString(4,cerFileName);
                ps.setString(5,bojFileName);
                ps.setString(6,isicFileName);
                ps.setString(7,loanFileName);
```

```
                int status = ps.executeUpdate();
                if (status > 0) {
                    System.out.println("File uploaded successfully...");
                    System.out.println("Uploaded Path: " + uploadPath);
                    response.sendRedirect("index2.jsp");
                }
            } catch (SQLException e) {
                out.println("Exception: " + e);
                System.out.println("Exception1: " + e);
            } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(UploadComReg.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
            } finally {
                try {
                    if (ps != null) {
                        ps.close();
                    }
                    if (JOconnection.conn != null) {
                        JOconnection.conn.close();
                    }
                } catch (SQLException e) {
                    out.println(e);
                }
            }

        } catch (IOException | ServletException e) {
            out.println("Exception: " + e);
            System.out.println("Exception2: " + e);
        }
    }

}
package Servlet;

import OOP.JOconnection;
import java.io.IOException;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.MultipartConfig;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
//Upload file to mySQL code from
https://github.com/Java-Hub18/Servlet-JSP/tree/11ba5e0f8d56be4e8a847ec3d4df1f99623ae
2c0/ServletUploadDownloadFile

@WebServlet(name = "UploadCompany", urlPatterns = {"/UploadCompany"})
@MultipartConfig(fileSizeThreshold = 1024 * 1024 * 10, // 10 MB
        maxFileSize = 1024 * 1024 * 1000, // 1 GB
        maxRequestSize = 1024 * 1024 * 1000)          // 1 GB
public class UploadCompany extends HttpServlet {

    PreparedStatement ps = null;
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/plain;charset=UTF-8");
        try {

            String name = request.getParameter("name");
            String employee = request.getParameter("employee");
            String phoneNum = request.getParameter("phoneNum");

            try {
                JOconnection.JConnect();
                String sql = "insert into company(name, employeeInCharge, phoneNumber)
values(?, ?, ?)";
                ps=JOconnection.conn.prepareStatement(sql);
                        ps.setString(1,name);
                ps.setString(2,employee);
                ps.setString(3,phoneNum);

                int status = ps.executeUpdate();
                if (status > 0) {
                    response.sendRedirect("index2.jsp");
                }
            } catch (SQLException e) {
                System.out.println("Exception1: " + e);
            } catch (ClassNotFoundException ex) {
```

```
java.util.logging.Logger.getLogger(UploadCompany.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
            } finally {
                try {
                    if (ps != null) {
                        ps.close();
                    }
                    if (JOconnection.conn != null) {
                        JOconnection.conn.close();
                    }
                } catch (SQLException e) {
                    System.out.println(e);
                }
            }

        } catch (IOException e) {
            System.out.println("Exception2: " + e);
        }
    }

}
package Servlet;

import OOP.JOconnection;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.nio.file.StandardCopyOption;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.MultipartConfig;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import jakarta.servlet.http.Part;
//Upload file to mySQL code from
https://github.com/Java-Hub18/Servlet-JSP/tree/11ba5e0f8d56be4e8a847ec3d4df1f99623ae
2c0/ServletUploadDownloadFile

@WebServlet(name = "UploadStatement", urlPatterns = {"/UploadStatement"})
@MultipartConfig(fileSizeThreshold = 1024 * 1024 * 10, // 10 MB
    maxFileSize = 1024 * 1024 * 1000, // 1 GB
    maxRequestSize = 1024 * 1024 * 1000)      // 1 GB
public class UploadStatement extends HttpServlet {

  PrintWriter out = null;
  PreparedStatement ps = null;

  @Override
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    response.setContentType("text/plain;charset=UTF-8");
    try {
      out = response.getWriter();
      request.getSession(false);
      String folderName = "Files";
      String uploadPath = request.getServletContext().getRealPath("") + File.separator +
folderName;
      File dir = new File(uploadPath);
      if (!dir.exists()) {
        dir.mkdirs();
      }
      Part stex = request.getPart("statementExcel");
      Part stpdf = request.getPart("statementPDF");
      Part vat = request.getPart("VAT");
      Part pnd = request.getPart("PND");
      Part profit = request.getPart("profit");
      String company = request.getParameter("Company");
      int year = Integer.parseInt(request.getParameter("Year"));

      String stexFileName = stex.getSubmittedFileName();
      String stpdfFileName = stpdf.getSubmittedFileName();
      String vatFileName = vat.getSubmittedFileName();
      String pndFileName = pnd.getSubmittedFileName();
      String profitFileName = profit.getSubmittedFileName();

      InputStream isstex = stex.getInputStream();
```

```
        InputStream isstpdf = stpdf.getInputStream();
        InputStream isvat = vat.getInputStream();
        InputStream ispnd = pnd.getInputStream();
        InputStream isprofit = profit.getInputStream();

        Files.copy(isstex, Paths.get(uploadPath + File.separator + stexFileName),
StandardCopyOption.REPLACE_EXISTING);
        Files.copy(isstpdf, Paths.get(uploadPath + File.separator + stpdfFileName),
StandardCopyOption.REPLACE_EXISTING);
        Files.copy(isvat, Paths.get(uploadPath + File.separator + vatFileName),
StandardCopyOption.REPLACE_EXISTING);
        Files.copy(ispnd, Paths.get(uploadPath + File.separator + pndFileName),
StandardCopyOption.REPLACE_EXISTING);
        Files.copy(isprofit, Paths.get(uploadPath + File.separator + profitFileName),
StandardCopyOption.REPLACE_EXISTING);

        try {
           JOconnection.JConnect();
           String sql = "insert into financialstatement(company, year, excelStatement,
pdfStatement, vatSummary, pnd50, excelProfitAndInterest) values(?, ?, ?, ?, ?, ?, ?)";
           ps=JOconnection.conn.prepareStatement(sql);
                ps.setString(1,company);
           ps.setInt(2,year);
           ps.setString(3,stexFileName);
           ps.setString(4,stpdfFileName);
           ps.setString(5,vatFileName);
           ps.setString(6,pndFileName);
           ps.setString(7,profitFileName);
```

```
                int status = ps.executeUpdate();
                if (status > 0) {
                    System.out.println("File uploaded successfully...");
                    System.out.println("Uploaded Path: " + uploadPath);
                    response.sendRedirect("index2.jsp");
                }
            } catch (SQLException e) {
                out.println("Exception: " + e);
                System.out.println("Exception1: " + e);
            } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(UploadStatement.class.getName()).log(java.util.logging.L
evel.SEVERE, null, ex);
            } finally {
                try {
                    if (ps != null) {
                        ps.close();
                    }
                    if (JOconnection.conn != null) {
                        JOconnection.conn.close();
                    }
                } catch (SQLException e) {
                    out.println(e);
                }
            }

        } catch (IOException | ServletException e) {
            out.println("Exception: " + e);
            System.out.println("Exception2: " + e);
        }
    }

}
package Servlet;

// Code partly generated by ChatGPT

import OOP.JOconnection;
import java.io.IOException;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet(name = "UserEdit", urlPatterns = {"/UserEdit"})
public class UserEdit extends HttpServlet {

    PreparedStatement ps = null;

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
```

```java
response.setContentType("text/html;charset=UTF-8");

try {

    String Name = request.getParameter("username");
    String Access = request.getParameter("access");

    try {
        JOconnection.JConnect();

        // Update the user information
        String sql = "UPDATE register SET access = ? WHERE username = ?";
        ps = JOconnection.conn.prepareStatement(sql);
        ps.setString(1, Access);
        ps.setString(2, Name);

        int status = ps.executeUpdate();
        if (status > 0) {
            System.out.println("Updated");
            System.out.println(Name);
            System.out.println(Access);
            response.sendRedirect("User.jsp");
        } else {
            System.out.println("not update");
            System.out.println(Name);
            System.out.println(Access);
        }
    } catch (SQLException e) {
            System.out.println("Exception1: " + e);
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(UploadCompany.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } finally {
            try {
                if (ps != null) {
                    ps.close();
                }
                if (JOconnection.conn != null) {
                    JOconnection.conn.close();
                }
            } catch (SQLException e) {
                System.out.println(e);
            }
        }

    } catch (IOException e) {
        System.out.println("Exception2: " + e);
    }
  }
}

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
this license
```

```
 * Click nbfs://nbhost/SystemFileSystem/Templates/JSP_Servlet/Servlet.java to edit this
template
 */
package Servlet;
//Download file from mySQL code from
https://github.com/Java-Hub18/Servlet-JSP/tree/11ba5e0f8d56be4e8a847ec3d4df1f99623ae
2c0/ServletUploadDownloadFile

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.OutputStream;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

/**
 *
 * @author IngIng
 */
@WebServlet("/getFiles")
public class getFiles extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public getFiles(){
        super();
```

```
}


public static int BUFFER_SIZE = 1024 * 100;
public static final String UPLOAD_DIR = "Files";
public static String fileName = null;


@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
  /**
   * *** Get The Absolute Path Of The File To Be Downloaded ****
   */
  fileName = request.getParameter("fileName");
  if (fileName == null || fileName.equals("")) {
    /**
     * *** Set Response Content Type ****
     */
    response.setContentType("text/html");


    /**
     * *** Print The Response ****
     */
    response.getWriter().println("<h3>File " + fileName + " Is Not Present .....!</h3>");
  } else {
    String applicationPath = getServletContext().getRealPath("");
    String downloadPath = applicationPath + File.separator + UPLOAD_DIR;
    String filePath = downloadPath + File.separator + fileName;
    System.out.println(fileName);
    System.out.println(filePath);
    System.out.println("fileName:" + fileName);
    System.out.println("filePath :" + filePath);
    File file = new File(filePath);
    OutputStream outStream = null;
    FileInputStream inputStream = null;

    if (file.exists()) {

        /**
         * ** Setting The Content Attributes For The Response Object
         * ***
         */
        String mimeType = "application/octet-stream";
        response.setContentType(mimeType);

        /**
         * ** Setting The Headers For The Response Object ***
         */
        String headerKey = "Content-Disposition";
        String headerValue = String.format("attachment; filename=\"%s\"", file.getName());
        response.setHeader(headerKey, headerValue);

        try {

            /**
             * ** Get The Output Stream Of The Response ***
             */
```

```
                    outStream = response.getOutputStream();
                    inputStream = new FileInputStream(file);
                    byte[] buffer = new byte[BUFFER_SIZE];
                    int bytesRead = -1;

                    /**
                     * ** Write Each Byte Of Data Read From The Input Stream
                     * Write Each Byte Of Data Read From The Input Stream Into
                     * The Output Stream ***
                     */
                    while ((bytesRead = inputStream.read(buffer)) != -1) {
                        outStream.write(buffer, 0, bytesRead);
                    }
                } catch (IOException ioExObj) {
                    System.out.println("Exception While Performing The I/O Operation?= " +
ioExObj.getMessage());
                } finally {
                    if (inputStream != null) {
                        inputStream.close();
                    }

                    outStream.flush();
                    if (outStream != null) {
                        outStream.close();
                    }
                }
            } else {
                /**
                 * *** Set Response Content Type ****
                 */
                response.setContentType("text/html");

                /**
                 * *** Print The Response ****
                 */
                response.getWriter().println("<h3>File " + fileName + " Is Not Present .....!</h3>");
            }

        }
    }
}
```